THE UNIVERSITY OF
MEMPHIS.
Computer Science

Dreamers. Thinkers. Doers.

Final Technical Report

Game Theoretic Approaches to Protect Cyberspace

For

The Office of Naval Research (ONR)
Grant N00014-09-1-0752

By

Sajjan Shiva
Principal Investigator

Dipankar Dasgupta and Qishi Wu
Co-Principal Investigators

Sankardas Roy
Postdoctoral Researcher

Team Members
Harkeerat Bedi, Vivek Datla, Charles Ellis, Nisrine Enyinda,
Beata Kubiak, Vivek Shandilya, and Chris Simmons

Department of Computer Science
University of Memphis
Memphis, TN, USA

Technical Report No. CS-10-001
April 30, 2010

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
30-04-2010	Final	April, 2009 - April, 2010

4. TITLE AND SUBTITLE

Game Theoretic Approaches to Protect Cyberspace

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014-09-1-0752

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Sajjan Shiva, Dipankar Dasgupta, Qishi Wu

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Memphis,
Office of Research Support
Services, Administration 315
Memphis, TN 38152-3370

8. PERFORMING ORGANIZATION REPORT NUMBER
CS-10-001

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
875 North Randolph Street, Arlington
VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Unlimited

13. SUPPLEMENTARY NOTES
-

14. ABSTRACT

The area of cyberspace defense mechanism design has received immense attention from the research community for more than two decades. However, the cyberspace security problem is far from completely solved. In this project we explored the applicability of game theoretic approaches to address some of the challenging cyber security issues: (a) We built a state-of-the-art attack taxonomy which can provide the system administrator with information on how to mitigate or remediate an attack. (b) We conducted a thorough survey of the existing game-theoretic solutions to cyber security problems and proposed a detailed taxonomy, which points out that this area requires more attention from the research community. (c) We proposed stochastic game models for generic cyber activities (attacks and defenses), which eliminate the unrealistic assumptions of the existing models. We validated the effectiveness of our model via extensive simulation. (d) We modeled the interaction between a class of attacks (such as the Denial of Service (DoS) and Distributed Denial of Service (DDoS)) and the possible countermeasures as a two-player general-sum game. We validated our analytical results via simulation experiments. (e) We compiled a set of metrics which can evaluate the cost and benefit of a game-theoretic defense solution. In addition, we have proposed a Game Theory Inspired Defense Architecture (GIDA).

15. SUBJECT TERMS
Cyber security, game theory, stochastic games, imperfect information, static and dynamic games, general-sum games, performance metrics, security metrics.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Sajjan Shiva
U	U	U	UU	86	19b. TELEPHONE NUMBER *(include area code)* 901-678-5465

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Abstract

The area of cyberspace defense mechanism design has received immense attention from the research community for more than two decades. However, the cyberspace security problem is far from completely solved. In this project we explored the applicability of game theoretic approaches to address some of the challenging cyber security issues: (a) We built a state-of-the-art attack taxonomy which can provide the system administrator with information on how to mitigate or remediate an attack. (b) We conducted a thorough survey of the existing game-theoretic solutions to cyber security problems and proposed a detailed taxonomy, which points out that this area requires more attention from the research community. (c) We proposed stochastic game models for generic cyber activities (attacks and defenses), which eliminate the unrealistic assumptions of the existing models. We validated the effectiveness of our model via extensive simulation. (d) We modeled the interaction between a class of attacks (such as the Denial of Service (DoS) and Distributed Denial of Service (DDoS)) and the possible countermeasures as a two-player general-sum game. We validated our analytical results via simulation experiments. (e) We compiled a set of metrics which can evaluate the cost and benefit of a game-theoretic defense solution. In addition, we have proposed a Game Theory Inspired Defense Architecture (GIIDA).

Keywords: Cyber security, game theory, stochastic games, imperfect information, static games, dynamic games, general-sum games, performance metrics, security metrics.

Publications resulted from this grant: The research accomplishments of this project have been reported in the following 5 publications.

(a) "AVOIDIT: A Cyber Attack Taxonomy"; S. Shiva, C. Simmons, C. Ellis, S. Roy, D. Dasgupta, and Q. Wu; Technical Report: CS-09-003, University of Memphis, August, 2009.

(b) "A Survey of Game Theory as Applied to Network Security"; S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya and Q. Wu; Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS), 2010.

(c) "A Stochastic Game Model with Imperfect Information"; S. Shiva, S. Roy, H. Bedi, D. Dasgupta, and Q. Wu; Proceedings of the 5th International Conference on Information Warfare and Security (ICIW), 2010.

(d) "On Modeling and Simulation of Game Theory-based Defense Mechanisms against DoS and DDoS Attacks"; Q. Wu, S. Shiva, S. Roy, C. Ellis, and V. Datla; Proceedings of the 43rd Annual Simulation Symposium (ANSS'10) in the Spring Simulation Multiconference (SpringSim), 2010. Received the Best Paper award at ANSS'10 and the Overall Best Paper award at SpringSim'10.

(e) "Game Theory for Cyber Security"; S. Shiva, S. Roy, and D. Dasgupta; Proceedings of the 6th Annual Cyber Security and Information Intelligence Research Workshop (CSIIRW), ORNL, 2010.

Contents

1 Introduction

National Security Presidential Directive 54 defines cyber space as the interdependent network of information technology infrastructures, and includes the Internet, telecommunication networks, computer systems and embedded processors and controllers in critical industries [74]. The Nation's economic progress and social well-being are becoming increasingly dependent on cyberspace. On the other hand, the growing inter-connectivity and the increasing availability of the computational power for the attacker is providing for distributed and sophisticated attacks [34]. Attackers can disrupt telecommunications, electrical power, energy pipelines, refineries, financial networks and other critical infrastructures [73]. Recent incidents indicate that cyber attacks can cause significant damage to governments, private enterprises, and the general public in terms of money, data confidentiality, and reputation [92, 30, 88].

The research community has been paying attention to the network security problem for more than two decades. However, the problem is far from being completely solved. We frequently see a race between the security specialists and the attackers in the following sense: one day an intelligent solution is proposed to fix a network vulnerability, and the next day the attackers come up with a smarter way to circumvent the proposed countermeasure. The most important factor which makes this problem difficult is that the local network, which needs to be secured, is typically connected to the Internet and major parts of the Internet are beyond the control of network administrators. However, the Internet has become an integral component of running the daily business of government, financial institutions, and the general public. As a result, there is a pressing need to design countermeasures for network attacks.

Traditionally, network security solutions employ either protective devices such as firewalls or reactive devices such as Intrusion Detection Systems (IDSs) and both of them are used in conjunction. The intrusion detection algorithms are either based on identifying an attack signature or detecting the anomalous behavior of the system. Once an attack is detected the employed IDS notifies the network administrator who then takes an action to stop or mitigate the attack. However, currently IDSs are not very sophisticated and they rely on ad-hoc schemes and experimental work. The current IDS technology may prove sufficient for defending against casual attackers using well known techniques, but there is still a need to design tools to defend against sophisticated and well organized adversaries.

The weakness of the traditional network security solutions is that they lack a quantitative decision framework. To this end, a few groups of researchers have started advocating the utilization of game theoretic approaches. As game theory deals with problems where multiple players with contradictory objectives compete with each other, it can provide us with a mathematical framework for analysis and modeling network security problems. As an example, a network administrator and an attacker can be viewed as two competing players participating in a game. In addition, game theory has the capability of examining hundreds of thousands of possible scenarios before taking the best action; hence, it can sophisticate the decision process of the network administrator to a large extent. As a result, several game theoretic approaches have recently been proposed to address network security issues.

In this project we explored the applicability of game theoretic approaches to address some of the challenging

cyber security issues. We conducted a survey of the literature related to the cyber attacks techniques to be better aware of the problem space, and then performed extensive research on how game theory can enrich the solution space.

We built a state-of-the-art attack taxonomy which classifies attacks by attack vectors, operational impact, defense, informational impact, and target. In addition to enhancing our knowledge about the various attack methodologies, this taxonomy can provide the system administrator with information of how to mitigate or remediate a particular attack.

In addition, we conducted a thorough survey of the existing game-theoretic solutions to cyber security problems and proposed a detailed taxonomy for classifying them. Highlighting the basic game type used in the defense mechanisms, while abstracting detailed differences, this taxonomy provides the reader with a global view of the current solution space. We observe that this area requires more attention from the research community.

Further, we observe that the prior stochastic game models for network security assumed that players have perfect information, which is a strong assumption. We propose a stochastic game model for generic cyber activities (attacks and defenses), which eliminates the unrealistic assumptions of the existing models. We validated the effectiveness of our model via extensive simulation in MATLAB. In addition, prior body of work related to network security did not present an algorithm to compute the equilibrium of a general-sum stochastic game. We explored the game theory literature to find such an algorithm and discovered that only recently game theoreticians proposed one such algorithm. We analyzed this algorithm, verified the analytical results via simulation, and found that this algorithm has many limitations.

Furthermore, we focus on a particular class of attacks, namely the DoS/DDoS attacks and model the interaction between these attacks and the possible countermeasures as a two-player non-zero-sum game. We study the existence of the Nash equilibrium which represents the smartest strategy of the players. We also show the benefit of using the game-theoretic defense mechanism for the network administrator. We validated our analytical results via simulation experiments using NS-3. We created a new module in NS-3 for our experiment, which we call as NetHook. NetHook provides the means for an application or module to have direct access to packets as they traverse the Internet Stack.

We also propose a set of metrics which can evaluate the cost and benefit of a game-theoretic defense solution. We define metrics that allow us to evaluate the security level, performance, and quality in a game theoretic defense architecture. We discuss how to compare different game theoretical defense models based on these metrics. We note that the ranking of a game model may change over time due to the dynamic nature of the cyber scenarios.

In addition, we propose a few important research directions for future work. In particular, we envision a game inspired defense architecture (GIDA) which leverages a game theoretic model to counter cyber attacks. GIDA will be capable of transparently observing network traffic, identifying malicious activity, measuring the risk, and acting upon that information in a way that will offer the best defense measure for that situation. The brain of GIDA is a game model which decides the best countermeasure after a thorough analysis of the cost and reward.

Below we summarize our contributions through this project:

- We analyzed the state-of-the-art in attack technology, and compiled a vulnerability-centric cyber attack taxonomy. Our taxonomy intends to provide a defender with vulnerability information which encompasses the attack, the impact of the attack on a targeted system, and the corresponding countermeasure. This taxonomy was published as a technical report by the Computer Science Department at University of Memphis [86].

- We performed a thorough survey of the literature related to the application of game theory to address network security problems. We proposed a complete and detailed taxonomy of the current game-theoretic approaches. Our survey paper was presented at the 43rd Hawaii International Conference on System Sciences [80].

- We extended prior stochastic game models by relaxing the assumption that the players have perfect information about the current state of the system. We evaluate the equilibrium condition in our model through theoretical analysis and detailed simulation. Our paper on this work was presented at the 5th International Conference on Information Warfare and Security [84]. We also analyzed the only algorithm available in the literature to compute a general-sum stochastic game, verified the analytical results via simulation, and found that this algorithm has many limitations.

- We propose game models to capture the interaction between the DoS/DDoS attacks and the potential mitigation techniques. We also present the theoretical analysis for the attacker's and defender's strategy which can lead to the Nash equilibrium. We validate our analytical results via extensive simulation in NS-3. We implemented a new module in NS-3, NetHook, that provides packet inspection capabilities similar to that of Linux NetFilter. Our paper on this work appeared in the Simulation Multiconference (SpringSim), 2010 [97].

- We compiled a set of metrics which can evaluate the cost and benefit of a game-theoretic defense solution. We define metrics that allow us to evaluate the security level, performance, and quality in a game theoretic defense architecture. We discuss how to compare different game theoretical defense models based on these metrics.

Organization of the rest of this report is as follows. In Section 2 we discuss our research on attack taxonomy. In Section 3 we provide a brief overview of game theory to set the background of our other research directions. We present our survey on game theoretic security solutions in Section 4. Next, we discuss our research on stochastic game models in Section 5, and we present our game models for DoS attacks in Section 6. Finally, we discuss our research on metrics in Section 7, and conclude in Section 8.

2 AVOIDIT: A Cyber Attack Taxonomy

Cyber attacks have created a global threat, both in defending local and global networks. Attacks are becoming more sophisticated and possess the ability to spread to numerous vulnerable hosts in a matter of seconds [78]. It is essential to provide tools necessary in detecting, classifying, and defending from various types of attacks. A variety of taxonomies aim at classifying vulnerabilities or attacks, but to date they have limitations in providing a defense strategy that can be used in a local application setting. This can be due to the enormous possibilities of defense strategies. We believe that coupling a defense mechanism with an attack taxonomy would enable a network administrator to not only understand the vulnerability, but also the strategy needed to mitigate and/or remediate the potential exploitation. Limitations exist toward providing defense strategies within an attack taxonomy. This presents an invaluable research area focused on the information a network administrator can apply when attempting to defend the network against cyber attacks. We propose a solution that addresses the shortcomings of existing taxonomies.

There is a deficient standard when disseminating vulnerability information, making it difficult for analysis with multiple vulnerabilities for potential defense. Landwehr et al. [49] and Lindqvist et al [52] state a taxonomy is most useful when it classifies threats in scope that correspond to potential defenses. This taxonomy differs from previous taxonomies, as it aids a defender to not only identify attacks, but also defense measures to mitigate and remediate attack vulnerabilities. One approach to gaining insight into attacker's target is to consider the attack paths, or combination of exploits [70]. AVOIDIT intends to provide a defender with vulnerability details to what encompasses an attack and any impact the attack may have on a targeted system. A blended attack exploits one or more vulnerabilities to perform an attack against a target [64]. AVOIDIT is able to classify blended attacks by providing the ability to label various vulnerabilities of an attack in a tree-like structure.

People question the impact a cyber attack has once its target is compromised. AVOIDIT provides useful information to the network administrator. We provide a mean to classify vulnerabilities that lead to cyber attacks with methods to mitigate and remediate vulnerabilities to help alleviate the impact of a successful exploitation. Avoiding the attack could simply require defending against propagation or further damage once an attack is identified. In order to better grasp this scenario, we provide several representative examples of attacks and how our proposed taxonomy successfully classifies well known attacks with defensive strategies.

2.1 A Brief Survey of Attack Taxonomies

Kjaerland [48] proposed a taxonomy of cyber-intrusions from Computer Emergency Response Team (CERT) related to computer crime profiling, highlighting cyber-criminals and victims. In this research, attacks were analyzed using facet theory and multidimensional scaling (MDS) with Method of Operation, Target, Source, and Impact. Each facet contains a number of elements with an exhaustive description. Kjaerland uses these facets to compare commercial versus government incidents. Kjaerland's taxonomy focuses on the motive of the attacker in an attempt to quantify why the attack takes place, and where the attack originated. Her taxonomy contains some limitations as she provides a high level view to the methods of operation without

providing more details to the methods that can be used in identifying attack inception.

Hansman and Hunt [40] proposed a taxonomy with four unique dimensions that provide a holistic classification covering network and computer attacks. Their taxonomy provides assistance in improving computer and network security as well as consistency in language with attack description. The first dimension being attack vector is used to classify the attack. The second dimension classifies the target of the attack. The third dimension consists of the vulnerability classification number, or criteria from Howard's taxonomy [42]. The fourth and final dimension highlights the payload or effects involved. Within each dimension various levels of information are provided to supply attack details. Hansman et al. mentioned the need of future work to improve classifying blended attacks, which is a limitation within their taxonomy. Another limitation is the lack of vulnerability information, which prohibits capturing information to aid in protecting a system from attacks.

Chakrabarti et al. [19] proposed a taxonomy focused on the Internet and its infrastructure as the basis for highlighting attacks and security. Infrastructure attacks can lead to considerable destruction due to different Internet infrastructure components having various trust relationships with one another. Chakrabarti et al. proposed a taxonomy consisting of four categories on Internet infrastructure attacks: DNS hacking, Route table poisoning, Packet mistreatment, and Denial of Service. They used the categories to develop a comprehensive understanding of the security threats. Chakrabarti et al. presented a valid point in securing the Internet infrastructure and provides techniques for securing the infrastructure. Their taxonomy is limited to layers one through three and lacks a comprehensive list of ways an infrastructure can be attacked, including the possibility of blended attacks. With limited research performed on aiding Internet infrastructure security, Chakrabarti et al. provided new research development in this area.

Mirkovic and Reiher [62] offer a comprehensive taxonomy of Distributed Denial of Services (DDoS) attack and defense mechanisms in aim to classify attacks and defense strategies. This research highlight features of attack strategies, where the strategies are imperative in devising countermeasures. Mirkovic and Reiher's taxonomy of DDoS attacks is categorized by Degree of Automation, Exploited Weakness, Source Address Validity, Attack Rate Dynamics, Possibility of Characterization, Persistent Agent Set, Victim Type, and Impact on Victim. These categories are used to examine the exploitation, the victim impact, and characteristics with exploiting a DDoS attack. In addition to classifying DDoS attacks, Mirkovic and Reiher developed a taxonomy of DDoS defenses consisting of Activity Level, Cooperation Degree, and Deployment Location. The combination classifying DDoS attacks and defenses within a taxonomy provides communication of threats to foster cooperation between researchers for discussing solutions.

Lough [57] proposed an attack-centric taxonomy called VERDICT (Validation Exposure Randomness Deallocation Improper Conditions Taxonomy). Lough focuses on four major causes of security errors: Improper Validation, Improper Exposure, Improper Randomness, and Improper Deallocation. He labels these four characteristics with a prefix of "Improper" with attacks being thought of as improper conditions. Validation refers to improperly validating or unconstrained data, which also includes physical security. Exposure involves the improper exposure of information that could be used directly or indirectly for the exploitation of a vulnerability. Randomness deals with the fundamentals of cryptography and the improper usage of ran-

domness. Deallocation is the improper destruction of information, or residuals of data, which also includes dumpster diving. He uses one or more of these characteristics to describe vulnerability within a system. Hansman and Hunt [40] describe Lough's taxonomy as lacking pertinent information that would be beneficial for knowledge bodies, such as CERT, to classify day-to-day attacks and issuing advisories. Lough's taxonomy lacks the classification to the type of attack, such as worms, Trojans, viruses, etc.

Howard [42] provides an incident taxonomy that classifies attacks by events, which is an attack directed at a specific target intended to result in a changed state. The event involves the action and the target. He highlights all steps that encompass an attack and how an attack develops. The attack consists of five logical steps an attacker performs to achieve an unauthorized result. Those steps are: tools, vulnerability, action, target, and unauthorized result. The tool refers to the mechanism used to perform the attack; the vulnerability is the type of exploit used to perform attack. The action refers to the method used by the attacker to perform the attack (i.e. Probe, Scan, Authenticate, etc.). The target is the intention the attack is attempting to compromise, and the unauthorized result is the change state caused due to the attack. Although Howard presents a useful taxonomy that provides an informative baseline for cyber intrusions, he lacks the details needed for thorough insight into the attack.

2.2 Our Proposed Taxonomy: AVOIDIT

A taxonomy defines what data is to be recorded and how like and unlike samplings are to be distinguished [49]. In developing a successful taxonomy, there are requirements that should be observed for universal acceptance. In Section 2 we analyze previous taxonomies and highlight valuable aspects that are needed to create a complete useful taxonomy [57, 42]. These requirements include the following:

Accepted – builds on previous work that is well accepted.

Mutually exclusive – each attack can only be classified into one category, which prevents overlapping.

Comprehensible – clear and concise information; able to be understood by experts and those less familiar.

Complete/exhaustive – available categories are exhaustive within each classification, it is assumed to be complete.

Unambiguous – involves clearly defined classes, with no doubt of which class an attack belongs.

Repeatable – the classification of attack should be repeatable.

Terms well defined – categories should be well defined, and those terms should consist of established terminology that is compliant within the security community

Useful – the ability to be used and gain insight into a particular field of study, particularly those having great interest within the field of study.

Applying these requirements for a complete taxonomy, we propose AVOIDIT. The AVOIDIT taxonomy provides, through application, a knowledge repository used by a defender to classify vulnerabilities that an attacker can use. Fig. 1 provides an overview of our proposed taxonomy, which provides details to support comprehending each attack classification and how a variety of attacks are represented in each category.

2.2.1 Classification by Attack Vector

When an attack takes place, there is a possibility it uses several vectors as a path to a full blown cyber attack. An attack vector is defined as a path by which an attacker can gain access to a host. This definition includes vulnerabilities, as it may require several vulnerabilities to launch a successful attack. In this section we list several vulnerabilities that are used to render a majority of attacks.

1. Misconfiguration - An attacker can use a configuration flaw within a particular application to gain access to a network or personal computer to cause a variety of attacks. Settings that are improperly configured, usually default settings, are an easy target for an attacker to exploit [83].

2. Kernel Flaws - An attacker can use a kernel flaw within an operating system, which is the core code of an operating system, to gain certain privileges to exploit a vulnerability within the operating system.

3. Buffer Overflow - Buffer overflow is caused when a piece of code does not adequately check for appropriate input length and the input value is not the size the program expects. Cowan [23] describes a buffer overflow when a buffer with weak or no bounds checking is populated with user supplied data. An attack can exploit a buffer overflow vulnerability leading to a possible exploitation of arbitrary code execution, often of privileges at the administrative level with the program running [83]. Buffer Overflow can occur in both stack and heap memory locations. A buffer overflow constitute majority of attacks [23]. A heap buffer overflow occurs in the heap data area, which is dynamically allocated by the application running [40] .

4. Insufficient Input Validation - A program fails to validate the input sent to the program from a user [83]. An attacker can exploit an insufficient input validation vulnerability and inject arbitrary code, which commonly occurs within web applications.

5. Symbolic Links - A file that points to another file [83]. An attacker can exploit a symbolic link vulnerability to point to a target file for which an operating system process has write permissions.

6. File Descriptor - A file that uses numbers from a system to keep track of files, as opposed to file names [83]. Exploitation of a file descriptor vulnerability allows an attacker the possibility of gaining elevated privileges to program related files.

7. Race Condition - Occurs when a program attempts to run a process and the object changes concurrently between repeated references [9]. An exploitation of race condition vulnerabilities allows an attacker to gain elevated privileges while a program or process is in privilege mode [83].

8. Incorrect File/Directory Permission - An incorrect permission associated to a file or directory consists of not appropriately assigning users and processes [83]. Exploiting this vulnerability can allow a multitude of attacks to occur.

9. Social Engineering - The process of using social interactions to acquire information about a victim, and or their computer system [17]. These types of attacks provide quick alternatives in disclosing information to assist an attack that in normal circumstances may not be available.

2.2.2 Classification by Operational Impact

Classification by Operational Impact involves the ability for an attack to culminate and provide high level information known by security experts, as well those less familiar with cyber attacks. We provide a mutually exclusive list of operational impacts that can be categorized and concisely presented to the public.

1. Misuse of Resources - An unauthorized use of IT resources [48]. We can extend this definition to consider any IT related function that require a certain privilege and those privileges are converted into an abusive action.

2. User Compromise - A perpetrator gaining unauthorized use of user privileges on a host, as a user compromise [48].

3. Root Compromise - Gaining unauthorized privileges of an administrator on a particular host [48]. We shall extend this notion slightly by including any elevated privileges above a normal user including administrative and/or root level privileges to a particular system.

4. Web Compromise - A website or web application using vulnerabilities to further an attack [48]. An attack can occur through a web compromise, usually via cross site scripting or sql injection.

5. Installed Malware - By exploiting some vulnerability an attack can be launched via user installed malware, whether user installed or drive-by installation. Provos et al. [78] discussed the implications of installed malware allowing the adversary to gain full control of the compromised systems leading to the ex-filtration of sensitive information or installation of utilities that facilitate remote control of the host.

6. Virus - A form of installed malware, where Hansman and Hunt [40] describes a virus as a piece of code that will attach itself through some form of infected files, which will self-replicate upon execution of program. Types of viruses include boot record infectors, file infectors, and macros.

7. Spyware - A type of malware program that is covertly installed and infects its target by collecting information from a computing system without owner's consent [37, 27]. The collected information is commonly used by attackers for financial gain, either identity theft or email marketing [37].

8. Trojan - A benign program to the user that allows unauthorized backdoor access to a compromised system [94]. A trojan is a common way to introduce a victim into a multitude of attacks.

9. Worms – A self-replicating computer program. Worms do not require human intervention to propagate as it is a self-replicating program that spreads throughout the network [65]. Worms include mass mailing and network aware worms.

10. Arbitrary Code Execution - Involves a malicious entity that gains control through some vulnerability injecting its own code to perform any operation the overall application has permission [26].

11. Denial of Service - Denial of Service (DoS) is an attack to deny a victim access to a particular resource or service [18], and has become one of the major threats and rated among the hardest Internet security issues [26]. In this section we will provide details into the types of DoS attacks.

12. Host Based - A Host based DoS aims at attacking a specific computer target within the configuration, operating system, or software of a host. These types of attacks usually involved resource hogs, aimed at consuming up all resources on a computer; crashers, which attempts to crash the host system [40].

13. Network Based - A Network based DoS targets a complete network of computers to prevent the network of providing normal services [26]. Network based DoS usually occur in the form of flooding with packets [40], where the network's connectivity and bandwidth are the target [26].

14. Distributed - A distributed denial of service uses multiple attack vectors to obtain its goal [62]. A Distributed Denial of Service (DDoS) is becoming more popular as an attacker's choice of DoS.

2.2.3 Classification by Defense

We extend previous attack taxonomy research to include a defense classification. Killourhy, et al. [47] state an attack taxonomy should be able to help the defender. In this section we highlight several strategies a defender can employ to remain vigilant in defending against attacks. We provide the possibility of using both mitigation and remediation when classifying attack defenses, as an attack could be first mitigated before a remediation can occur.

1. Mitigation - Prior to vulnerability exploitation or during an attack, there are several steps a defender can use to minimize damage an attack has caused, or has the potential to cause. An example can involve an installation of a worm that propagate over the network, one instance could be to remove a set of hosts from the network and route traffic, while the administrator works on removal of the worm. Mitigation involves lessening the severity of the attack.

2. Remove from Network - The ability of an administrator to remove infected hosts preventing further damage. As the example described above, a particular worm may reside in a network and begins propagation.

3. Whitelisting - Whitelisting involves a list of permissible connections that are known to the defender. An attack could be directed at a particular software, which may reside on predetermined port.

4. Reference Advisement - Notes provided by the defender to mitigate an attack, or a vulnerability/vendor database reference number used to alleviate a vulnerability or attack.

5. Remediation - In the presence or prior to vulnerability exploitation, there are resolution steps that are available to a defender to prevent an attack. Remediation would involve taking the appropriate steps to correct the situation prior to or during an exploitation.

6. Patch System - Applying patches the vendor has released due to some vulnerability within software in use. When a vulnerability or attack is present, on various cases, a defender fails to utilize the patches a vendor provides.

7. Correct Code - Steps within an organization to release a code patch to a specific application that will close the potential for an attacker to exploit.

2.2.4 Classification by Informational Impact

An attack on a targeted system has potential to impact sensitive information in various ways. Huttchinson [43] state information is the power and weapons at all strategic, tactic, and informational levels. A committed resource must be able defend information warfare strategies in an effort to protect themselves against theft, disruption, distortion, denial of service, or destruction of sensitive information assets [25]. In this section we classify an attacks impact, or the effect on information and define the criteria used.

1. Distort - A distortion in information, usually when an attack has caused a modification of a file. When an attack involves distort, it is a change to data within a file, or modification of information from the victim [48].

2. Disrupt - A disruption in services, usually from a Denial of Service. When an attack involves disrupt, it is an access change, or removal of access to victim or to information [48].

3. Destruct – A destruction of information, usually when an attack has caused a deletion of files or removal of access. Destruct is the most malicious impact, as it involves the file deletion, or removal of information from the victim [48].

4. Disclosure - A disclosure of information, usually providing an attacker with a view of information they would normally not have access to. Kjaerland [48] describes disclosure as unauthorized disclosure of information, with the possibility of leading to other compromises.

5. Discovery – To discover information not previously known. For example, when a scanning tool probes for information, the information discovered can be used to launch an attack on a particular target.

2.2.5 Classification by Attack Target

Various attacks target a variety of hosts, leaving the defender unknowingly susceptible to the next attack.

1. Operating System (Kernel / User / Driver) - Responsible for the coordination of activities and the sharing of resources of a computer. An attack can be formulated to target vulnerabilities within a particular operating system.

2. Network - Target a particular network or gain access through a vulnerability within a network or one of the network protocols [40].

3. Local - An attack targeting a user's local computer.

4. User - An attack against a user is an attack to retrieve a user's personal information.

5. Application – An attack towards specific software. An application can be either client or server. A client application is software that is available to aid a user performing common tasks. A server application is software designed to serve as a host to multiple concurrent users.

2.3 Taxonomy Comparison

In this section we use the taxonomies described above to compare AVOIDIT with past computer attacks and vulnerabilities. This section will highlight how our cyber attack taxonomy successfully captures vulnerability attack information and provide a defender with countermeasures that can be efficient in preventing or assuaging successful attacks.

2.3.1 SQL Slammer

This section provides details into the SQL Slammer worm. Slammer was able to perform 55 million scans per second and compromised ninety percent of vulnerable hosts in 10 minutes [64]. Table 1 classifies the SQL Slammer worm.

In Table 1, Lough's taxonomy is too general to provide useful information in describing the attack; Howard's taxonomy provides preliminary information. Hansman and Hunt's taxonomy is able capture more detail in comparison to Howard. Our taxonomy provides information on what caused the worm infection, and possible defense strategies a network administrator can use to reduce the malware's ability to further propagate and cause damage. Using AVOIDIT, if the first insertion was alleviated, the Slammer worm would not be able to spread.

2.3.2 Microsoft RPC Stack Overflow

In 2008, a Windows Server service Remote Procedure Call (RPC) stack buffer overflow vulnerability [91, 76] was exploited and is currently "in the wild". This RPC service provides print support and network pipe sharing were other users were able to access services over a network. The notable Conficker or Downadup attacks use these vulnerabilities to perform attacks on vulnerable systems. Table 2 classifies the RPC buffer overflow.

Classifying the buffer overflow vulnerability using Lough or Howard's taxonomy, we are unable to view the details, and unable to aid in defending against the vulnerability exploit. Using Hansman and Hunt's taxonomy, we may have been able to classify the attack, but the variations of the vulnerability the various attacks exploited are not present. With this particular vulnerability exploitation, you can view AVOIDIT

Table 1: SQL Slammer Attack Classification: Comparison of taxonomic classifications for the SQL Slammer worm.

LOUGH

Name	Improper Validation	Improper Exposure	Improper Randomness	Improper Deallocation
Slammer	X	X		

HOWARD

Name	Tools	Vulnerability	Action	Target	Unauthorized Result
Slammer	Script	Configuration, Design	Prob, Modify	Network	Corruption of Information

HANSMAN

Name	1st Dimension	2nd Dimension	3rd Dimension	4th Dimension
Slammer	Network Aware Worm	MS SQL Server 2000	CAN-2002-0649	Stack Buffer Overflow & UDP packet flooding DoS

AVOIDIT

Name	Attack Vector	Operational Impact	Informational Impact	Defense	Target
Slammer	Misconfiguration	Installed Malware: Network Aware Worm	Discovery	Mitigation: Whitelisting CAN-2002-0649	Network
Slammer	Buffer Overflow	Installed Malware: Network Aware Worm	Distort	Remediation: Patch System	Application

as being able to thoroughly classify the vulnerability, potential blended attacks, and attack variations that specifically exploited the Windows buffer overflow vulnerability.

2.4 AVOIDIT Classification Structure

In this section we were able to classify a multitude of vulnerabilities and attacks. AVOIDIT benefits from the ability of being able to classify attacks in a tree-like structure, providing the ability to classify the allusive blended attack. Predecessors [40, 57] state that providing a tree-like structure is a solution to solving the blended attack, but claim this particular structure can become unorganized. We provide our taxonomy in a tree-like structure to successfully classify common vulnerabilities and cyber attacks to provide defenders with the needed information to defend their networks. Table 3 provides insight into how a searchable schema can be obtain we classify attacks using a tree-like structure, which enable a searchable schema. By using a parent-child relationship, AVOIDIT is able to display how multi-staged attacks can be captured, classified, and disseminated.

Table 2: MS RPC Stack Overflow Classification: A comparison of taxonomic classification for the Microsoft Remote Procedure Call (RPC) Overflow attack.

LOUGH

Name	Improper Validation	Improper Exposure	Improper Randomness	Improper Deallocation
MS RPC Stack Overflow	X	X		

HOWARD

Name	Tools	Vulnerability	Action	Target	Unauthorized Result
MS RPC Stack Overflow	Script	Design	Modify	Process	Increased Access

HANSMAN

Name	1st Dimension	2nd Dimension	3rd Dimension	4th Dimension
MS RPC Stack Overflow	Stack Buffer Overflow	Windows Server	CVE-2008-4250	Corruption of Information

AVOIDIT

Name	Attack Vector	Operational Impact	Informational Impact	Defense	Target
MS RPC Stack Overflow	Buffer Overflow: Stack	Installed Malware:ACE	Distort	Mitigation: RA: VU#827267 Remediation: Patch System	OS: Windows Server
Gimmiv.A	Buffer Overflow: Stack	Installed Malware:Trojan	Disclosure	Mitigation: RA: Microsoft Remediation: Patch System	OS: Windows Server
Conficker	Buffer Overflow: Stack	Installed Malware:Worm	Disrupt	Mitigation: RA: Microsoft Remediation: Patch System	OS: Windows Server, 2000, XP

2.5 AVOIDIT Applied in a Network

In this section we show how AVOIDIT can be used within cyber security to support a defender against malicious attackers.

AVOIDIT is intended to be used in multiple aspects of a network defense policy. It can be used to store event notifications within a database to educate administrators of attack frequency. The network administrator can also use an AVOIDIT organized knowledge repository in order to locate strategies that are appropriate for securing their network against vulnerabilities that can be exploited and used for unauthorized access. AVOIDIT used in a network defense strategy can improve the overall level of security. Our taxonomy can be used by applications that can offer a multitude of functions. The most obvious of these is that the taxonomy can be used to provide a defender with information related to the commonality, frequency, and vendor response pertaining to an event in which a vulnerability was exploited. This information will then be used to identify and implement defense measures. Previous taxonomies in Section 2 lack the structure of useful

Table 3: AVOIDIT attack classification structure for multiple attacks, including multi-stage blended attacks.

ID	Parent	Name	Attack Vector	Operational Impact	Defense	Informational Impact	Target
001		Slammer	Misconfig	Malware: Network Aware Worm	Mitigation: Whitelisting Remediation: Patch System	Discovery	Network
002	001	Slammer	Buffer Overflow	Malware: Network Aware Worm	Remediation: Patch System	Distort	Application
003		Zotob	Buffer Overflow	Malware: Worm	Remediation: Patch System	Distort	OS
004	003	Zotob	Buffer Overflow	Malware: Worm	Remediation: Patch System	Distort	Local
008		SamyXSS	Design Flaw	Web Compromise	Remediation: Correct Code	Disrupt	User
009		Debian Admin	Kernel Flaw	Root Compromise	Remediation: Patch System	Disclosure	OS
010	009	Debian Admin	Kernel Flaw	DoS	Mitigation: RA	Distort	OS
011		Yamanner	Social Engineering	Web Compromise	Mitigation: RA	Disclosure	Application: Server: Email
012	011	Yamanner	Design Flaw	Malware: Mailing Worm	Mitigation: RA	Disrupt	User
013		MS RPC Overflow	Buffer Overflow	Malware: ACE	Mitigation: RA: VU#827267 Remediation: Patch System	Distort	OS: Windows Server
014	013	Gimmiv.A	Buffer Overflow	Malware: Trojan	Mitigation: RA: Microsoft Remediation: Patch System	Disclosure	OS: Windows Server
015	013	Conficker	Buffer Overflow	Malware: Worm	Mitigation: RA: Microsoft Remediation: Patch System	Disrupt	OS: Windows Server 2000, XP

information to classify attacks through vulnerabilities that can be used in an application to assist a defender against an attack. Our taxonomy provides a more apparent approach to educate the defender on possible cyber attacks using vulnerability details. AVOIDIT will be used in a future game theoretic defense system to capture vulnerability information to provide a network administrator with a solution when defending against cyber attacks [29]. Until now, previous attack taxonomies have not been applied in a defense model, thus through application, our taxonomy presents a better approach in capturing and disseminating valuable information in defending a network against cyber attacks.

2.6 AVOIDIT Limitations

Attacks have become increasingly present in the cyber world, and being able to provide the ability to prevent all attacks is extremely difficult. In this section we will highlight some of the limitations of AVOIDIT.

2.6.1 Lack of Defense Strategies

The defense strategies in our taxonomy present a defender with an appropriate starting point to mitigate and/or remediate an attack. The plausible defenses are enormous, so the proposed taxonomy provides a high level approach to cyber defense. Although AVOIDIT is extensible, more research is needed to provide an exhaustive list of possible defense strategies for each vulnerability exploited.

2.6.2 Physical Attack Omission

Physical attacks are an important aspect in achieving security. While it is necessary to understand physical attacks, our proposed taxonomy focuses on cyber attacks. Further research can be done to include the physical aspect of cyber security, which may include the end hosts of an attack.

2.6.3 Summary

This section introduces a cyber attack taxonomy that enhances the cyber security industry. AVOIDIT will classify attacks by attack vectors, operational impact, defense, informational impact, and target. This classification scheme will aid a defender in protecting their network by providing vital attack information. It is presented in a tree-like structure to neatly classify common vulnerabilities used to launch cyber attacks.

We are aware of the possibility of new attack manifestation, therefore AVOIDIT could be extended to include new categories within each classification. AVOIDIT will provide a defender with the appropriate information to make an educated decision in defending against cyber attacks. Creative approaches to defending attacks will become available and providing an extensible taxonomy able to capture new defenses is imperative to defense. We believe AVOIDIT provides a foundation for the cyber security community and provide the ability to continuously grow as attacks and defenses become more sophisticated. In future work, to build a Game Theoretic Defense System, we will investigate the applicability of AVOIDIT in determining the action space of the attacker [29].

3 Basics of Game Theory

This section identifies the premise of game theory to aid the understanding of the games referred later in this report. For a detailed introduction to game theory refer *A Course in Game Theory* [71]. Game theory describes multi-person decision scenarios as games where each player chooses actions which result in the best possible rewards for self, while anticipating the rational actions from other players.

A player is the basic entity of a game who makes decisions and then performs actions. A game is a precise description of the strategic interaction that includes the constraints of, and payoffs for, actions that the players can take, but says nothing about what actions they actually take. A *solution concept* is a systematic description of how the game will be played by employing the best possible strategies and what the outcomes might be.

The *consequence function* associates a *consequence* with each action the decision makers take. A *preference relation* is a complete relation on the set of consequences which model the preference of each player in the game. A *strategy* for a player is a complete plan of actions in all possible situations throughout the game. If the strategy specifies to take a unique action in a situation then it is called a *pure strategy*. If the plan specifies a probability distribution for all possible actions in a situation then the strategy is referred to as a *mixed strategy*.

A Nash equilibrium is a solution concept that describes a steady state condition of the game; no player would prefer to change his strategy as that would lower his payoffs given that all other players are adhering to the prescribed strategy. This solution concept only specifies the steady state but does not specify how that steady state is reached in the game. The Nash equilibrium is the most famous equilibrium, even though there are many other solution concepts used occasionally. This information will be used to define games that have relevant features for representing network security problems.

3.1 Definitions

Game

A description of the strategic interaction between opposing, or co-operating, interests where the constraints and payoff for actions are taken into consideration.

Player

A basic entity in a game that is tasked with making choices for actions. A player can represent a person, machine, or group of persons within a game.

Action

An action constitutes a move in the given game.

Payoff

The positive or negative reward to a player for a given action within the game.

Strategy

Plan of aetion within the game that a given player ean take during game play.

Perfect Information Game

A game in whieh eaeh player is aware of the moves of all other players that have already taken plaee. Examples of perfeet information games are: ehess, tie-tae-toe, and go. A game where at least one player is not aware of the moves of at least one other player that have taken plaee is ealled an imperfect information game.

Complete Information Game

This is a game in whieh every player knows both the strueture of the game and the objeetive funetions of all players in the game, but not necessarily the aetions. This term is often eonfused with that of perfeet information games but is distinct in the fact that it does not take into account the aetions eaeh player have already taken. Ineomplete information games are those in whieh at least one player is unaware of the strueture of the game or the objeetive funetion for at least one of the other players.

Bayesian Game

A game in whieh information about the strategies and payoff for other players is ineomplete and a player assigns a 'type' to other players at the onset of the game. Sueh games are labeled Bayesian games due to the use of Bayesian analysis in predieting the outeome.

Static/Strategic Game

A one-shot game in whieh eaeh player ehooses his plan of aetion and all players' deeisions are made simultaneously. This means when choosing a plan of action eaeh player is not informed of the plan of aetion ehosen by any other player. In the rest of this article, this elass of game is referred to as 'statie game'.

Dynamic/Extensive Game

A game with more than one stages in eaeh of whieh the players ean eonsider their aetion [71]. It ean be eonsidered as a sequential structure of the deeision making problems eneountered by the players in a static game. The sequences of the game can be either finite, or infinite. In the rest of this artiele, this elass of game is referred to as 'dynamic game'.

Stochastic Game

A game that involves *probabilistic transitions* through several states of the system. The game progresses as a sequence of states. The game begins with a start state; the players choose actions and reeeives a payoff that depend on the eurrent state of the game, and then the game transitions into a new state with a probability based upon players' aetions and the eurrent state.

4 Game Theory as Applied to Network Security

The weakness of the traditional network security solutions is that they lack a quantitative decision framework. To this end, a few groups of researchers have started advocating the utilization of game theoretic approaches. As game theory deals with problems where multiple players with contradictory objectives compete with each other, it can provide us with a mathematical framework for analysis and modeling network security problems. As an example, a network administrator and an attacker can be viewed as two competing players participating in a game. In addition, game theory has the capability of examining hundreds of thousands of possible scenarios before taking the best action; hence, it can sophisticate the decision process of the network administrator to a large extent. As a result, several game theoretic approaches have recently been proposed to address network security issues.

We survey the existing game theoretic solutions which are designed to enhance network security and present a taxonomy for classifying them. Highlighting the basic game type used in the defense mechanisms, while abstracting detailed differences, this taxonomy provides the reader with a global view of the problem and solution space. We do not advocate any specific defense game, rather the main purpose is to provide the reader with the current solution possibilities.

The rest of this section is organized as follows. Section 4.1 explains how network security problems can be modeled as a game. Section 4.2 classifies the current state of research and proposes a taxonomy. Finally, Section 4.3 and 4.4 highlight the differences between this report and other surveys in the field, and provide a summary.

4.1 Information Warfare as a Game

Global networks continue to undergo dramatic changes resulting in ever-increasing network size, interconnectivity, and accessibility, and a consequent increase in its vulnerability. Several recent Federal policy documents have emphasized the importance of cyber security to the welfare of modern society [14, 22]. The President's National Strategy to Secure Cyber Space [14] describes the priorities for response, reduction of threats and vulnerabilities, awareness and training, and national security and international cooperation. *Cyber Security: A Crisis of Prioritization* [22] describes the need for certain technologies for cyber security. Security should be an integral part of advanced hardware and software from the beginning, as described by Sun Microsystems, Cisco Systems, and Microsoft at the 2006 RSA Conference.

Next-generation information infrastructure must robustly provide end-to-end connectivity among computers, mobile devices, wireless sensors, instruments, etc. Cyber-security is an essential component of information and telecommunications, which impacts all of the other critical US infrastructures [31]. However, traditional cyber-security methods involve a never-ending cycle of detection and response to new vulnerabilities and threats. It is recognized that this patches-on-patches approach is a short fix and attests to the failure of the present cyber-security paradigm, and points to the need for a new and bold approach. The US-CERT [92] web site has currently more than 20,000 vulnerabilities (increasing by 50 to 60 per month), implying a world-wide cost more than 1 trillion dollar. The open web application security project also lists

top ten vulnerabilities of the year for web-based applications. "Build Security In" (BSI) [72] is a project of the Strategic Initiatives Branch of the National Cyber Security Division (NCSD) of the US Department of Homeland Security is for use by software developers, who want information and practical guidance on producing secure and reliable software. NSA has an effort on high-assurance computing platforms. The Trusted Computing Group [36] has an ongoing effort. Microsoft has an effort on next-generation secure computing [60].

In future warfare, cyberspace will play a major role where no one is guaranteed to have information dominance in terms of intelligence and accessibility. As a result, a game-theoretic approach of collaboration (carrot) and compelling (counter-) moves (stick) need to be played efficiently. This notion is not unlike the mutually assured destruction (MAD) of nuclear warfare. The question then becomes: How do we construct such a game theoretic approach in cyberspace?

In general, a game-theoretic approach works with at least two players. A player's success in making choices depends on the choices of others. In game theory, players are pitted against each other taking turns sequentially to maximize their gain in an attempt to achieve their ultimate goal [1]. In the field of cyber security, game theory has been used to capture the nature of cyber conflict. The attacker's decision strategies are closely related to those by the defender and vice versa. Cyber-security then is modeled by at least two intelligent agents interacting in an attempt to maximize their intended objectives.

Different techniques available in game theory can be utilized to perform tactical analysis of the options of cyber threat produced either by a single attacker or by an organized group. A key concept of game theory is the ability to examine the huge number of possible threat scenarios in the cyber system [38, 39]. Game theory can also provide methods for suggesting several probable actions along with the predicted outcome to control future threats. Computers can analyze all of the combinations and permutations to find exceptions in general rules, in contrast to humans who are very prone to overlooking possibilities. This approach allows identification of the what-if scenarios, which the human analyst may not have considered.

4.2 Taxonomy: Classification of Current Research

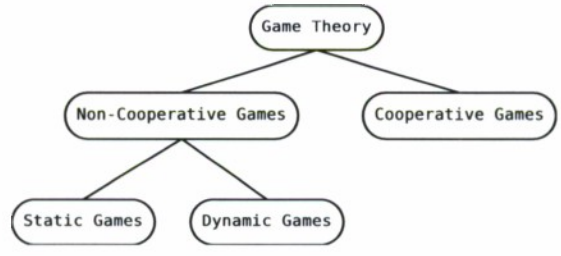

Figure 1: Classification of games

Figure 1 illustrates the basic classification of game theory. The existing game-theoretic research as applied to network security falls under non-cooperative games. As such, this report does not further expand upon 'cooperative games'. Figure 2 illustrates the classification of static games and lists the existing research works (related to network security) falling under each class. Figure 3 does the same for dynamic games.

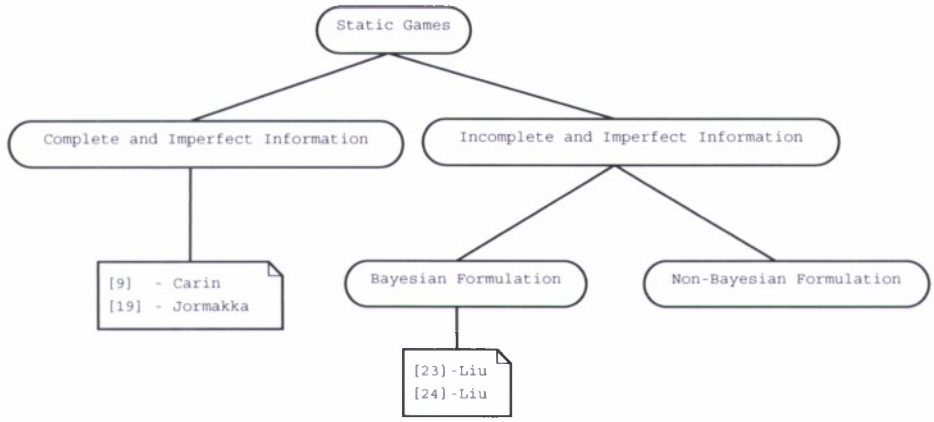

Figure 2: Classification of Static Games: Each rectangular leaf node lists the research works which fall under the corresponding category. Each research work is represented by the reference number and the first author name.

Section 4.2.1 discusses existing works involving static games while Section 4.2.2 deals with existing works involving dynamic games. Section 4.2.3 discusses a few other works which do not directly fall under these classes. Finally, Section 4.2.4 presents some directions for future research.

4.2.1 Static games

Since a static game is a one-shot game, by definition all static games are of imperfect information. According to the completeness of information, static games can be classified into two sub-classes as listed below. We briefly discuss the existing research works which fall under each sub-class of static games.

4.2.1.1 Complete imperfect information

Jormokka et al. [45] introduced a few examples of static games with complete information where each example represents an information warfare scenario. For each scenario the authors found the best strategy of the players in a quantitative form. In particular, they investigated if more than one Nash equilibria exist and if so, then which one is most likely to appear as the outcome given the players' strategies. These examples show that depending on the scenario the players could get the benefit of a bold strategy or a mixed strategy.

Carin et al. [15] presented a computational approach to quantitative risk assessment for investment efficient strategies in cyber security. The focus of this work was how to protect the critical intellectual property in private and public sectors assuming the possibility of reverse engineering attacks. The authors proposed an *attack/protect economic model* cast in a game theoretic context.

4.2.1.2 Incomplete imperfect information

Liu et al. [54] presented a methodology to model the interactions between a DDoS attacker and the network administrator. This approach observed that the ability to model and infer attacker intent, objectives, and strategies (AIOS) is important as it can lead to effective risk assessment and harm prediction. An *incentive-based* game-theoretic model to infer AIOS was discussed in this work. A few bandwidth parameters were

used as the metric to measure the impact of the attack and the countermeasure, which in turn measures the attacker's, and defender's, incentive. The work also observed that the best game model to choose depends on the degree of accuracy of the employed IDS and the degree of correlation among the attack steps. The work reported simulation results involving game plays following the Bayesian model while the simulation experiment was performed on ns-2. The topology considered in the simulation experiment consists of 64 source hosts connected to one victim machine via 4 levels of routers. Each router is capable of employing the *pushback* mechanism as part of the defense strategy. A set of Nash equilibrium strategies were computed via the simulation.

Liu et al. [56] focused on the intrusion detection problem in mobile ad-hoc networks. Their two-player game model is based on a Bayesian formulation and they analyzed the existence of Nash equilibria in static scenario. The defender updates his prior beliefs about the opponent based on new observations. This work investigated the Bayesian Nash Equilibria (BNE) in the static model. The authors also presented some results from the experiments performed on the ns-2 simulator.

4.2.2 Dynamic games

A dynamic game can be either of complete or incomplete information. Moreover, a dynamic game may involve perfect or imperfect information. So, there are four sub-classes of dynamic games as listed below. For each sub-class of dynamic games, we briefly discuss the existing research works which fall under the corresponding sub-class.

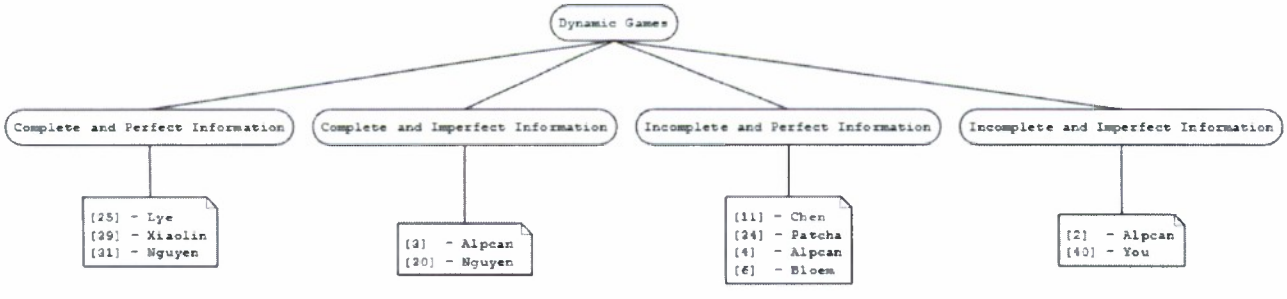

Figure 3: Classification of Dynamic Games: Each rectangular leaf node lists the research works which fall under the corresponding category. Each research work is represented by the reference number and the first author name.

4.2.2.1 Complete perfect information

Lye et al. [58] proposed a game model for the security of a computer network. In this work, an enterprise network was envisioned as a graph of 4 nodes (web server, file server, work station and external world) along with the traffic state for all the links. It is a two-player (administrator, attacker), stochastic, general-sum game and the authors focused on 3 attack scenarios namely, defaced website, denial-of-service, and stealing confidential data. The game was described from the point of view of both players. A formal model defined the game as a 7-tuple— the set of network states, the action set for each player, the state transition function, the reward function and a discount factor. In particular, this work considered a stochastic game

involving 18 network states and 3 actions for each player at each state. The state transition probabilities and the reward matrices are assigned using the domain knowledge. With different initial conditions a set of Nash Equilibria were calculated using a non-linear program in Matlab.

Xiaolin et al. [98] proposed a Markov game theory based model for risk assessment of network information system considering the security status of both present and future. They identified that threats acting on vulnerability can induce risk and the risk will be larger and larger by threat spreading. On the other hand, the risk will be smaller and smaller by the system administrator's repairing the vulnerability. Thus, they established a game of threats and vulnerabilities. Essentially, the experiment involves a game of complete and perfect information with two players. Authors formulated a function to capture the damage and used it to assess the risk. Using the damage function the system administrator would select the repair strategy which minimizes the maximum damage. To evaluate their model they constructed a risk assessment platform with four subsystems which are Malicious code Detection Subsystem, Vulnerability Detection Subsystem, Asset Detection Subsystem and Risk Assessment Subsystem. They used Trojan.Mybot-6307 as a threat, and three assets to define states. Their results are similar or better than the traditional assessment model like Fault Tree Analysis (FTA) because they effectively incorporated the potential risk also. They came up with a repair table of vulnerability states and threat states. They claimed that the model also leads to the best system repair scheme.

In Nguyen et al.'s [68] model, an attacker and the network administrator participate in a two-player zero-sum stochastic game. This work assumed that the network consists of a set of interdependent nodes whose security assets and vulnerabilities are correlated. It utilized the concept of linear influence networks [63] and modeled the interdependency among nodes by two weighted directed graphs, one signifying the relationship of security assets and the other denoting vulnerability correlation among the nodes. This research presented one numerical example considering a small network of three nodes to explain how to compute the optimal strategies of the players.

4.2.2.2 Complete imperfect information

Alpcan et al. [4] modeled the interaction between malicious attackers to a system and the IDS using a stochastic (Markov) game. They captured the operation of the IDS sensor system using a finite-state Markov chain, and considered three different information structures: (a) the players have full information about the sensor system characteristics and the opponents, (b) the attacker has no information about the sensor system characteristics, and (c) each player has only information about his own costs, past actions, and past states. A few illustrative examples and numerical analysis were presented for these three cases. Tools such as value iterations to solve Markov decision processes (MDP) [8], minimax-Q [53], and naive Q-learning [8] were used to find the best strategies of the players.

Nguyen et al. [67] viewed the network security problem as a sequence of nonzero-sum games played by an attacker and a defender. This game model, called 'fictitious play (FP)', conservatively considers that the players cannot make perfect observations of each other's previous actions. This work studied the impact of the error probabilities associated with the sensor system on the Nash equilibrium strategies of the players considering two scenarios— (a) each player is aware of these error probabilities, and (b) neither player

knows these error probabilities. Both classical and stochastic FP games are investigated via simulation.

4.2.2.3 Incomplete perfect information

Chen [20] in his doctoral dissertation used game theoretic model to design the response for the importance-scanning Internet worm attack. The main idea is that defenders can choose how to deploy an application, that is the group distribution, when it is introduced to Internet to minimize the worm propagation speed. The attacker can choose the optimal group scanning distribution to maximize the infection speed. Thus a game would be played between the attacker and the defender. The attacker should choose so as to maximize the minimum speed of worm propagation, while defender wants to minimize the maximum speed of worm propagation. By framing the problem this way it turns out to be a zero sum game and a min-max problem. The optimal solution for this problem is that defender should deploy the application uniformly in the entire IP-address space or in each enterprise network, so that the best strategy that the attacker exploits is equivalent to random scanning strategy. This work gave a game theoretical framework to design the locations of vulnerable and high value hosts over a network.

Patcha et al. [75] proposed a game theoretic approach to model intrusion detection in mobile ad-hoc networks. The authors viewed intrusion detection as a game played between the attacker node and the IDS hosted on the target node. The objective of the attacker is to send a malicious message with the intention of attacking the target node. The modeled game is a basic signaling game which falls under the domain of multi-stage dynamic non-cooperative game.

Alpcan et al. [5] investigated the problem of Nash Equilibrium Design for quite a general class of games from an optimization and control theoretic perspective. The work is theoretical and the analysis is general though aimed at information networks. They restricted their treatment to a class of games where players do not manipulate the game by deceiving the system designer and where utility functions accurately reflect user preferences. They further discussed the games with incomplete information with two objective functions: Quality of service (QoS)-based and utility maximization. They concluded that though the tragedy of commons or price of anarchy is unavoidable in pure games, it is circumvented altogether when additional mechanism such as "pricing" are included. They explored the pricing dynamics in different conditions. They inferred that "loss of efficiency" is not an inherent feature of a broad class of games with built-in pricing systems, but merely a misconception that often stems from arbitrary choice of game parameters. Finally, they give a brief overview of Nash Equilibrium dynamic control. They focused on how long does the game approach Nash equilibrium when many players are trying to solve it in a distributed way. They suggested a feedback control system approach with pricing as a control input to make the system robust and to control the system's progress and investigated system's controllability in general.

Bloem et al. [10] modeled intrusion response as a resource allocation problem based on game theory. A cost is associated with attacks and responses. This problem, including imperfections in the sensor outputs, was first modeled as a continuous game. The strategies are discretized both in time and intensity of actions, which eventually leads to a discretized model. The reaction functions uniquely minimize the strictly convex cost functions. After discretization, this becomes a constrained integer optimization problem. To solve this they introduced their dynamic algorithm, Automatic or Administrator Response algorithm (AOAR).

They classified attacks into those resembling previous attacks and those that do not, and many such intuitive classes with Kohonen self-organizing maps, a neural net, and the response cost is minimized. The simulations captured variation in vulnerability, value and cost of actions. Their results showed system performs improves after using AOAR.

Though majority of Liu et al.'s [54] approaches fall under static games with incomplete and imperfect information (Section 4.2.1.2), one of their approaches falls under this category.

4.2.2.4 Incomplete imperfect information

Alpcan et al. [3] modeled the interaction of an attacker and the network administrator as a repeated game with 'finite steps' or 'infinite steps'. This work assumed that the sensor system which is deployed to detect the attacks is imperfect and considered the sensor system as a third 'fictitious' player similar to the 'nature' player in standard game theory. It found the Nash equilibrium in a repeated game via simulation considering a simple scenario with three specific attacks. The Nash equilibrium strategies were computed assuming simple cost functions for the players.

You et al. [101] described how to model the network security scenario considering the interaction between the hacker and the defender as a two player, zero sum game. It gave a taxonomy of relevant game theory and network security terms and suggested a correlation between them. They pointed out at the utility of Nash and Bayesian Equilibria in representing the concepts to predict behavior and analyzed the interaction between the attacker and the defender. They gave a list of game theory terms that are relevant in the network security scenario and explained them. They explained how min max theorem for this game is formulated. They concluded by suggesting that to solve this problem linear algorithms would be appropriate.

The research reported in [4], [67] and [75] which are described under other classes of games also contain additional approaches that fall under this class of game.

4.2.3 Other work

Bursztein et al. [13] presented a model for evaluating the plausibility of successful attacks on a given network with interdependent files and services. This work provided a logic model that accounts for the time needed to attack, crash, or patch network systems. Rather than providing a game theoretic model, the work used the given time and topology constraints to determine if an attack, or defense, would be successful. The example presented described a high-availability web server configuration with interdependent elements and considered the strategic actions of the attacker as well as the defender.

Sun et al. [90] analyzed information security problem in the mobile electronic commerce chain. They claimed that the application of game theory in information safety is based on the hypothesis of player's perfect rationality, while in reality, the main body of information security only has the bounded rationality, which is just the assumption of Evolutionary Game theory. They introduced the penalty parameter in the problem if an organization in the mobile electronic commerce chain does not invest in information security. They calculated replicator dynamics of this game. They analyzed Evolutionary Stable strategy to get the results which formulate that the pay off to the organizations for investing is higher than not investing. This

is an application of evolutionary game theory to the investment strategy in the network security to obtain the best security pay off.

Sun et al. [89] used game theory to make the analysis and put forward strategy suggestions for defender organization to invest in information security. It is concerned about management and not the technology of the information security. They formulated the problem of two organizations investing in the security, with parameters such as for investment, security risk and disasters. They presented a pay off matrix. They did the Nash Equilibrium analysis for both pure and mixed strategy and showed them to be consistent. To make the investing a rational option they introduced a penalty parameter associated with not investing. They concluded by presenting an argument for encouraging organizations the investment in information security.

4.2.4 Discussion: scope of future research

Many of the current game-theoretic security approaches are based on either static game models [54, 56] or games with perfect information [58, 98, 5, 10] or games with complete information [68]. However, in reality a network administrator often faces a dynamic game with incomplete and imperfect information against the attacker. Some of the current models involving dynamic game with incomplete and imperfect information are specific to wireless networks [75] while a few others [3, 101] do not consider a realistic attack scenario.

In particular, some of the limitations of the present research are: (a) Current stochastic game models [58] only consider perfect information and assume that the defender is always able to detect attacks; (b) Current stochastic game models [58] assume that the state transition probabilities are fixed before the game starts and these probabilities can be computed from the domain knowledge and past statistics; (c) Current game models assume that the players' actions are synchronous, which is not always realistic; (d) Most models are not scalable with the size and complexity of the system under consideration.

4.3 Related work

This section briefly discusses the existing body of other research related to our survey topic, and mentions how the existing work differs from our work. It also discusses a few research works which focus on the taxonomy of network attacks and cyber incidents. It is to be noted that good understanding of the attack taxonomy is a prerequisite to design a countermeasure.

Hamilton et al. [39] outlined the areas of game theory which are relevant to information warfare. The paper analyzed a few scenarios suggesting several potential courses of actions (COA) with predicted outcomes and what-if scenarios. Alpha-beta, alpha-beta star, and beta pruning with min-max search are suggested approaches. Hill climbing algorithm was suggested for predicting the opponent moves. In the domain of checkers, a linear programming technique using pattern recognition was cited as finding the optimal weights in a followup pass after hill climbing. Automatic tuning of evaluation functions by the chess program, Deep-Blue is highlighted. They concluded with speculating about great possibilities in applying game theory to information warfare. Hamilton et al.'s work focusses on a motivating example to illustrate the use of game theory in network security problems while we provide a taxonomy of the existing game-theoretic solutions.

Hamilton et al. [38] identified the following seven challenges in applying game theory to the domain of information warfare: (i) There is a limited database of relevant games played by real players, (ii) Both the attacker and the defender can launch multiple moves simultaneously, (iii) Players can take as long as they want to make moves, (iv) The defender may not be able to correctly able to identify the end goal of the opponent, (v) At each step the flow of the game may change so that the known legal moves, both in number and kind, may change for each player, (vi) The defender may find it hard to keep track of any possible change in the opponents resources and also his end goals, (vii) It is hard to define precisely the timing for move and state updates. The authors expected that these challenges could be addressed with some non-trivial breakthroughs in the research. We investigate how the existing game-theoretic solutions meet some of the above challenges.

Kjaerland [48] introduced existing body of research work related to computer crime profiling and proposed a taxonomy of cyber-intrusions, which provides insight into cyber-criminals and victims. In this research, Kjaerland focused on reported cyber intrusions reported from CERT. These attacks were analyzed using facet theory and multidimensional scaling (MDS) with Method of Operation, Target, Source, and Impact. Each facet contains a number of elements, each is mutually exclusive and elements exhaustively describe the facet. Kjaerland concluded the paper with comparing the incidents of commercial versus government incidents.

Hansman and Hunt [40] proposed a taxonomy consisting of four unique dimensions that provide a holistic classification that covers network and computer attacks, providing assistance in improving computer and network security as well as consistency in language with attack description. The first dimension is attack vector, which is used to categorize the attack into an attack class. The second dimension allows for the classification of attack targets, which can be classified to specific targets (e.g., OS:Linux:RedHat6.0). The third dimension consists of the vulnerability classification and the attack uses (e.g., CVE/CERT). The fourth and final dimension highlight the potential payload or effects involved (e.g., File Deletion). Within each dimension various levels of information are provided to successfully classify and supply attack details. Hansman and Hunt provided examples to conclude the proposed taxonomy is general to categorize attacks and mentioned the nced of future work to improve classifying blended attacks. There are several research works,e.g. [46], [64], which study network attacks.

Chakrabarti et al. [19] focused on the Internet and its infrastructure as being the basis for highlighting attacks and security. Where majority of research focused on securing the data being transferred, this research discussed attacks on the infrastructure which can lead to considerable destruction due to different Internet infrastructure components having various trust relationships with one another. Chakrabarti et al. categorized possible Internet infrastructure attacks, identified attacks within each category, solutions within each category, and presented guidelines for less researched areas. In their taxonomy of attacks they provided four categories on Internet infrastructure attacks (DNS hacking, Route table poisoning, Packet mistreatment, and Denial of Service). They used the categories to develop a comprehensive understanding of the security threats.

Mirkovic and Reihner [62] presented a taxonomy of Distributed Denial of Services (DDoS) attack and

defense mechanisms in aim to classify attacks and defense strategies. This work highlighted attack commonalities and important features of attack strategies. These strategies are vital in dictating the design of countermeasures. With focus on DDoS attacks, Mirkovic and Reihner created a taxonomy to examine the exploitation, the characteristics, and the victim impact of the attack. The taxonomy of DDoS attacks was categorized by Degree of Automation, Exploited Weakness, Source Address Validity, Attack Rate Dynamics, Possibility of Characterization, Persistent Agent Set, Victim Type, and Impact on Victim. Highlighting challenges defending against DDoS attacks, Mirkovic and Reihner developed a taxonomy of DDoS defenses consisting of Activity Level, Cooperation Degree, and Deployment Location. Mirkovic and Reihner concluded with the proposed taxonomies to provide communication of threats and related countermeasures aiming to foster cooperation between researchers for discussing solutions.

4.4 Summary

Hackers activities have significantly increased in cyber space, and have been causing damage by exploiting weaknesses in information infrastructure. Considerable efforts are continuously being made by the research community for the last two decades to secure networks and associated devices. Recently, researchers have been exploring the applicability of game theoretic approaches to address cyber security problems and have proposed a handful of competing solutions. Game theory offers promising perspectives, insights, and models to address the ever changing security threats in cyber space. This survey highlights important game theoretic approaches and their applications to network security and outlines possible directions for future research. It is to be noted that classes in the taxonomy could be divided into more detailed levels. It is obvious that new classes may need to be introduced in the taxonomy after new defense mechanisms are proposed in the future.

5 Stochastic Game Models with Realistic Assumptions

Prior stochastic game models for network security assume that players have perfect information, which is a strong assumption. We design a stochastic game model in which players may have imperfect information. In addition, prior body of work related to network security does not present an algorithm to compute the equilibrium of a general-sum stochastic game. We explore the current game theory literature to find such an algorithm and discover that only recently game theoreticians have proposed one such algorithm. We analyze this algorithm, verify the analytical results via simulation, and find that this algorithm has many limitations. Below we report our findings along the above two research directions: First, in Section 5.1, we discuss our imperfect information stochastic game model, and then, in Section 5.2, we discuss our research towards solving a general-sum stochastic game.

5.1 Considering Imperfect Information

To model attacks and defense mechanisms, a stochastic game model has been proposed in the literature [58, 59, 4]. The state of the game probabilistically changes depending on actions taken by the players (i.e., type of attacks and defender's response) and the system configurations. During each state transition, each player gets a payoff or incurs some cost (negative payoff). Techniques exist by which a player can determine the best strategy to get the highest overall payoff considering all of the possible strategies of the adversary. Game theoreticians formulate the solution concept of a stochastic game by the notion of Nash equilibrium, and have already provided the analysis indicating the existence of the equilibrium [28].

As stated, the prior stochastic game models for network security [58, 59] assume that the players have perfect information about the current state of the game, which implies that the defender is always able to detect an attack and the attacker is always aware of the employed defense mechanism. In real systems, a player uses a sensor (e.g., the defender's sensor can be a part of the Intrusion Detection System (IDS)) to observe the current status of the system to decide the strategy. It is widely believed that no real sensor can perfectly read the environment, i.e., usually there is a non-zero error probability. So, in most cases, the above assumption about perfect information does not hold in real life.

Section 5.1 relaxes this assumption and designs a stochastic game model which is able to capture more realistic scenarios. It considers that a player knows the system's true state at a particular moment with some error probability, i.e., at any given point of time the true state and a player's perception can be potentially different. With this additional constraint of imperfect information, Section 5.1 computes the best strategy for a player considering other players' choice of possible strategies.

In particular, Section 5.1 presents a theoretical analysis by which the defender can compute his/her best strategy to reach the Nash equilibrium of a stochastic game assuming the defender's sensor is imperfect. It is implicit that the defender knows the error probability of his/her sensor and the players' objectives are directly opposite, i.e., it is a zero-sum game. Moreover, Section 5.1 shows that if the defender follows the strategy prescribed by the perfect information model, then the Nash equilibrium is not achieved and the attacker's payoff can be more. Our algorithm for computing the best strategy runs offline well before the

game is being played, i.e., our game analysis is static. Furthermore, our theoretical results are validated via simulation experiments in MATLAB.

The major contributions of Section 5.1 are summarized below:

- We present a static analysis of an imperfect information zero-sum stochastic game and compute the best strategy of the system administrator in realistic scenarios.
- Our analysis and simulation experiments illustrate that the system administrator will be betteroff if he/she takes our strategy compared to the scenario when he/she executes the strategy prescribed by the perfect information models.

The rest of Section 5.1 is organized as follows: Section 5.1.1 briefly presents the perfect information stochastic game model. Section 5.1.2 and 5.1.3 introduces our imperfect information stochastic game model and also provides analysis and simulation results. Section 5.1.4 discusses the related work, and Section 5.1.5 concludes Section 5.1.

5.1.1 Preliminaries: A Stochastic Game Model with Perfect Information

This section provides a brief overview of a stochastic game model as discussed elsewhere [58, 59]. For further details of the stochastic game model refer to [28].

Lye et al. model the interaction between the attacks and the defense actions as a two players' ($k = 1, 2$) game where player 1 is the attacker and player 2 is the system administrator [58, 59]. This infinite-horizon stochastic game model considers N states.

The stochastic game is represented by a tuple (S, A^1, A^2, Q, R^1, R^2, β) whose elements are defined below.

1. $S = \{\xi_1, \xi_2, \ldots, \xi_N\}$ is the set of states. A state represents the current status of the whole system under consideration.

2. $A^k = \{A^k{}_{\xi_1}, A^k{}_{\xi_2}, \ldots, A^k{}_{\xi_N}\}$, $k = 1, 2$ where $A^k{}_{\xi_j} = \{\alpha^k_{j1}, \alpha^k_{j2}, \ldots, \alpha^k_{jM^k}\}$ is the action set of player k at state ξ_j. It is assumed that $M^k = |A^k{}_{\xi_j}|$ for all $1 \leq j \leq N$.

3. The state transition probabilities are represented by the function $Q : S \times A^1 \times A^2 \times S \rightarrow [0\ 1]$ which maps a pair of states and a pair of actions to a real number between 0 and 1. As an example, $Q(\xi 1, \alpha 11, \alpha 21, \xi 2) = 0.3$ is interpreted as the probability of state transition from state $\xi 1$ to $\xi 2$ given that player 1 takes action $\alpha 11$ and player 2 takes action $\alpha 21$.

4. The reward of player k is dctermined by the function $R^k : S \times A^1 \times A^2 \rightarrow \mathbb{R}$ which maps a state and a pair of actions to a real number. As an example, $R^1(\xi 1, \alpha 11, \alpha 21) = 42$ is interpreted as the reward gained by the attacker at state $\xi 1$ given that attacker takes action $\alpha 11$ and player 2 takes action $\alpha 21$. Negative reward represents the cost incurred by a player.

5. β, $0 < \beta < 1$ is the discount factor for discounting future rewards to calculate the overall payoff of a player in this infinite horizon game.

We now define the stationary strategy of a player. Stationary strategy is one that remains constant over time. Let $\Omega^{\mathrm{n}} = \{ \mathrm{p} \in R^n \mid \sum_{i=1}^{n} p_i = 1,\ 0 \leq p_i \leq 1\}$ be the set of probability vectors of length n. Let the function $\pi^k : \mathrm{S} \rightarrow \Omega^{\mathrm{M_k}}$ denote the strategy for player k where

$$\pi^k(s) = [\, \pi^k(s,\ \alpha_1),\ \pi^k(s,\ \alpha_2),\ \ldots,\ \pi^k(s,\ \alpha_{M_k})],$$

while $\pi^k(s,\ \alpha_i)$ is the probability with which player k selects the action α_i in state s. If π^k is such that $\forall\ s, i,\ \pi^k(s,\ \alpha_i)$ is 0 or 1, then π^k is called a pure strategy. Otherwise, π^k is called a mixed strategy.

During each state transition, player k gets a reward (defined by the function R^k) or incurs some cost (negative reward). To compute the overall payoff of player k, we consider the future moves which will change the present state to next states giving future payoff to player k. The overall payoff is computed by discounting the future payoff using the discount factor β. Let ${v^k}_{\pi^1,\ \pi^2}(s)$ denote the expected overall payoff of player k when the game starts at state s while the strategy of player 1 is π^1 and the strategy of player 2 is π^2. Let the vector ${v^k}_{\pi^1,\ \pi^2}$ denote the aggregate payoff of player k, where ${v^k}_{\pi^1,\ \pi^2} = [\, {v^k}_{\pi^1,\ \pi^2}(\xi_1),\ {v^k}_{\pi^1,\ \pi^2}(\xi_2),\ \ldots,\ {v^k}_{\pi^1,\ \pi^2}(\xi_{\mathrm{N}})]$.

Each player has the goal to maximize his expected payoff. The Nash equilibrium of this game is defined to be a pair of strategies $(\pi^1_*,\ \pi^2_*)$ which simultaneously satisfy the following equations component-wise:

$${v^1}_{\pi^1_*,\ \pi^2_*} \geq {v^1}_{\pi^1,\ \pi^2_*}\ \forall\ \pi^1 \in \Omega^{\mathrm{M_1}}$$

$${v^2}_{\pi^1_*,\ \pi^2_*} \geq {v^2}_{\pi^1_*,\ \pi^2}\ \forall\ \pi^2 \in \Omega^{\mathrm{M_2}}$$

A stochastic game is called zero-sum if one player's reward at each state transition is equal and opposite of the other player's reward, i.e., for all i, j, m we have $R^1(\xi\mathrm{i},\ \alpha 1\mathrm{j},\ \alpha 2\mathrm{m}) = -\ R^2(\xi\mathrm{i},\ \alpha 1\mathrm{j},\ \alpha 2\mathrm{m})$. It implies that for every pair of strategies the overall payoff of the players are same and opposite, i.e.,

$$\forall\ \pi^1 \in \Omega^{\mathrm{M_1}},\ \pi^2 \in \Omega^{\mathrm{M_2}} \quad {v^1}_{\pi^1,\ \pi^2} = -\ {v^2}_{\pi^1,\ \pi^2}.$$

For a zero-sum stochastic game which has a Nash equilibrium,$(\pi^1_*,\ \pi^2_*)$, the *value* of the game is considered as ${v^1}_{\pi^1_*,\ \pi^2_*}(s_1)$ where s_1 is the start state. Let V denote the value of the game.

We can compute the Nash equilibrium strategy of the players for a zero-sum stochastic game through a static analysis (offline analysis) of the game using the algorithm discussed in [28]. The algorithm used is basically an iterative non-linear optimization technique.

5.1.2 Our Model with Imperfect Information

The above game model assumes that the players have perfect information about the current state of the game. Our model presented in this section relaxes this assumption. Section 5.1.2.1 presents our imperfect information stochastic game model. Section 5.1.2.2 presents a static analysis and Section 5.1.3 provides the

simulation results.

5.1.2.1 The Model

Our model is an extension of the prior model (Seetion 5.1.1) and considers that a player k ($k = 1, 2$) observes the game's true state at a partieular moment by an imperfeet sensor deviee. That means, player k can view ξ_j as any state in the information set $I^k_{\xi_j}$ with some probability where $I^k_{\xi_j} = \{ \xi_{j_1}, \xi_{j_2}, \ldots, \xi_{j_p}\}$ with ξ_j being an element of $I^k_{\xi_j}$. Compared to the perfect information model, player k's aetion spaee may beeome wider, i.e., player k may take an aetion whieh is allowed at a state $\xi_{j_i} \neq \xi_j$ belonging to the information set, $I^k_{\xi_j}$.

Let $B^k_{\xi_j}$ denote the set of possible aetions of player k when his/her information set is $I^k_{\xi_j}$. Then

$$B^k_{\xi_j} = \bigcup_{\xi_i \in I^k_{\xi_j}} A^k_{\xi_i},$$

where $A^k_{\xi_i}$ denotes the aetion set of player k when he/she is sure that the true current state is ξ_i. Below we formally define the outeome of player k's extended aetion set $B^k_{\xi_j}$, eompared to $A^k_{\xi_j}$ in the previous model, when the true state is ξ_j. If player k takes an action $\alpha^k \in B^k_{\xi_j}$ when the true state is ξ_j but α^k is not in $A^k_{\xi_j}$, then in terms of the influenee on state transition probability, α^k is equivalent to player k taking no action at state ξ_j. However, regarding the influence on player k's payoff α^k may not be equivalent to player k taking no aetion at state ξ_j depending upon the eost of the exeeution of α^k.

Formally, our model is represented by a tuple, $(S,\ I^1,\ I^2,\ E^1,\ E^2,\ A^1,\ A^2,\ B^1,\ B^2,\ Q,\ R^1,\ R^2,\ \beta)$ whose elements are defined below.

1. $S = \{\xi_1, \xi_2, \ldots, \xi_N\}$ is the set of states.
2. $I^k = \{I^k{}_{\xi_1}, I^k{}_{\xi_2}, \ldots, I^k{}_{\xi_N}\}$, $k = 1, 2$ where $I^k{}_{\xi_j}$ represents the information set of player k when the true state is ξ_j, i.e., $I^k{}_{\xi_j} = \{\xi_{j_1}, \xi_{j_2}, \ldots, \xi_{j_p}\}$ (where p is an arbitrary positive integer) with the eondition that $\xi_j \in I^k{}_{\xi_j}$.
3. $E^k = \{E^k{}_{\xi_1}, E^k{}_{\xi_2}, \ldots, E^k{}_{\xi_N}\}$, $k = 1, 2$ where the j-th set $E^k{}_{\xi_j}$ represents the error probabilities of k-th player's sensor at the true state ξ_j over the corresponding information set, $I^k{}_{\xi_j}$.
4. $A^k = \{A^k{}_{\xi_1}, A^k{}_{\xi_2}, \ldots, A^k{}_{\xi_N}\}$, $k = 1, 2$ where $A^k{}_{\xi_j} = \{ \alpha^k_{j_1}, \alpha^k_{j_2}, \ldots, \alpha^k_{j_{M^k}}\}$ is the aetion set of player k at state ξ_j.
5. $B^k = \{B^k{}_{\xi_1}, B^k{}_{\xi_2}, \ldots, B^k{}_{\xi_N}\}$, $k = 1, 2$ where $B^k{}_{\xi_j}$ represents the extended aetion set of player k at $I^k{}_{\xi_j}$. That means, $B^k{}_{\xi_j} = \bigcup_{\xi_i \in I^k_{\xi_j}} A^k_{\xi_i}$. By introdueing identieal actions we ean make $|B^k{}_{\xi_j}|$ same for all $1 \leq j \leq N$. Let $T^k = |B^k{}_{\xi_j}|$.
6. The state transition probabilities are represented by the funetion $Q : S \times B^1 \times B^2 \times S \to [0\ 1]$ which maps a pair of states and a pair of aetions to a real number between 0 and 1. Our model assumes that for any state $\xi_j$if player k takes an action $\alpha^k_i \in B^k{}_{\xi_j}$ while α^k_i does not belong to $A^k{}_{\xi_j}$, then $Q\left(\xi_{j_1}, \alpha^k_{i_1}, \alpha^l_{i_2}, \xi_{j_2}\right) = Q(\xi_{j_1}, nop, \alpha^l_{i_2}, \xi_{j_2})$ where l represents the other player.

7. The reward of player k is determined by the function $R^k : S \times B^1 \times B^2 \rightarrow \mathbb{R}$ which maps a state and a pair of actions to a real number.

8. β, $0 < \beta < 1$ is a discount factor for discounting future rewards in this infinite horizon game.

We redefine the strategy function π^{k} of the perfect information model for this imperfect information model as π^{k}: $\mathrm{S} \rightarrow \Omega^{\mathrm{T_k}}$ where $\pi^{\mathrm{k}}(\mathrm{s}) = [\, \pi^{\mathrm{k}}(\mathrm{s},\ \alpha_1), \pi^{\mathrm{k}}(\mathrm{s},\ \alpha_2), \ldots, \pi^{\mathrm{k}}(\mathrm{s},\ \alpha_{\mathrm{T_k}})\,]$. The definition of the payoff vector of player k $(v^k{}_{\pi^1,\ \pi^2})$ and the Nash equilibrium, $(\pi^1_*,\ \pi^2_*)$ are similarly extended.

One major difference of this model from the perfect information game is as follows: As player k's sensor is not perfect, when his/her strategy π^k is executed in the true sense, his/her observed strategy (referred to as apparent strategy in the rest of Section 5.1), $\pi^{k'}$ is different from π^k. We will illustrate this further in Section 5.1.2.2.

5.1.2.2 A Static Analysis for a Game with Two States

We now present a static analysis of our game model, by which a player can compute his/her best strategy offline. Only a zero-sum game is considered. This analysis considers the worst-case scenario from the defender's point of view. It is assumed that only the defender's sensor is erroneous while the attacker can perfectly observe the current state of the game. It is to be noted that our analysis can be easily extended to the case where the attacker's sensor is also imperfect. Furthermore, this analysis is restricted to a game of two states for the sake of simplicity. In the future work, this analysis will be extended for games with more than two states. We focus on the following game as illustrated in Figure 4.

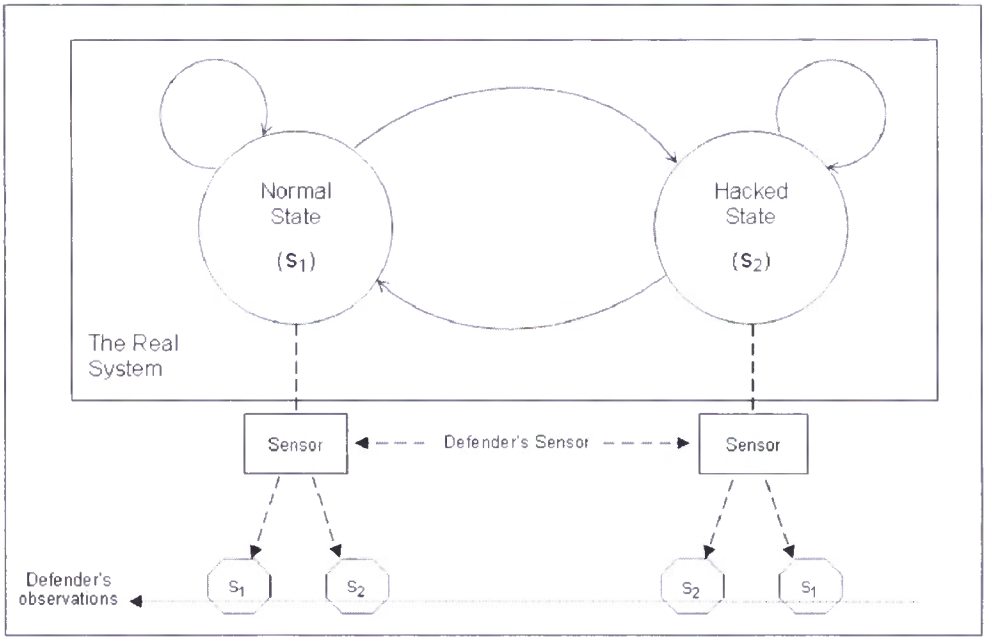

Figure 4: The state transition diagram and defender's observations — the same sensor is shown twice to indicate observations at different states

There are two states in this game. The system is either in *NormalState* (s_1) or in *HackedState* (s_2).The defender's sensor is imperfect and the error probability at state $s_1$and s_2 are γ_1 and γ_2, respectively. That means, when the true state is s_1, with probability γ_1 the defender observes that as s_2, and when the true state is s_2, the defender observes the state as s_1 with probability γ_2. However, it is assumed that the sensor's error probabilities (γ_1 and γ_2) are known to the defender. On the other hand, the attacker's sensor observes the current state with no error.

The action spaces of the players, A^1 and A^2 are as follows where a denotes 'attack', na denotes 'no attack', d denotes 'defense' and nd denotes 'no defense'. The first row in A^1 or A^2 represents the actions available in state $s_1$and the second row is for s_2.

$$A^1 = \begin{bmatrix} a & na \\ a & na \end{bmatrix} \quad and \quad A^2 = \begin{bmatrix} d & nd \\ d & nd \end{bmatrix}.$$

In this game, each player's extended action space (Section 5.1.2.1) remains same as the original action set. The strategy of the player k is represented by the probability distribution with which player k selects the available actions. The strategies of the players are represented by the following matrices π^1 and π^2:

$$\pi^1 = \begin{bmatrix} {\pi^1}_{11} & {\pi^1}_{12} \\ {\pi^1}_{21} & {\pi^1}_{22} \end{bmatrix} \quad and \quad \pi^2 = \begin{bmatrix} {\pi^2}_{11} & {\pi^2}_{12} \\ {\pi^2}_{21} & {\pi^2}_{22} \end{bmatrix}.$$

As an example, ${\pi^1}_{11}$ represents the probability with which player 1 selects action a and ${\pi^1}_{12}$ represents the probability with which player 1 selects action na at s_1 and ${\pi^1}_{11} + {\pi^1}_{12} = 1$.

State Occurrence Ratio (r_1,r_2): A stochastic game involves state transitions. The proportion of times a state s_i will occur during the whole play is called its occurrence ratio and is denoted by r_i. The value of $r_i$depends on the state transition probability function Q and the true strategies π^1 and π^2.

Given true strategies π^1 and π^2, we can compute the effective state transition probability matrix P whose dimension is $|S| \times |S|$. The element $P(i,j)$ represents the probability with which state s_i will switch to state s_j . Here, P is a 2×2 matrix.

We can compute $r_1$and $r_2$as follows. From basic theory of stochastic game [28] we know that $P^i(1,j)$ represents the probability that state s_j will occur at the ith transition.

$$r_1 = \lim_{n\to\infty} \frac{P(1,1) + P^2(1,1) + \ldots + P^n(1,1)}{n}$$

$$r_2 = \lim_{n\to\infty} \frac{P(1,2) + P^2(1,2) + \ldots + P^n(1,2)}{n}$$

As expected from the above two expressions we get $r_1 + r_2 = 1$. As the defender's sensor is not perfect, he/she can observe different occurrence ratios. As the attacker's sensor is perfect, now onwards the term 'apparent' only relates to the defender.

Apparent State Occurrence Ratio (r_1',r_2'): The apparent occurrence ratios of state s_1 and s_2 are as follows.

$$r_1' = (1-\gamma_1)\, r_1 + \gamma_2 r_2$$

$$r_2' = \gamma_1 r_1 + (1-\gamma_2)\, r_2$$

We stress the fact that the defender's true strategy, π^2 is different from his/her apparent strategy, $\pi^{2'}$, which he/she observes being executed. We represent $\pi^{2'}$ as follows.

$$\pi^{2'} = \begin{bmatrix} \pi^{2'}{}_{11} & \pi^{2'}{}_{12} \\ \pi^{2'}{}_{21} & \pi^{2'}{}_{22} \end{bmatrix}.$$

As an example, $\pi^{2'}{}_{11}$ represents the apparent probability of action d and $\pi^{2'}{}_{12}$ represents the apparent probability of action nd at s_1. Note that $\pi^{2'}{}_{11} + \pi^{2'}{}_{12} = 1$.

The defender's apparent strategy, $\pi^{2'}$is determined by his/her true strategy, π^2, sensor error probabilities (γ_1,γ_2) and the true state transition ratios, (r_1, r_2) as described in the following matrix equation. The matrix IIF is called the imperfect information factor and represents the influence of the sensor's errors.

$$\begin{bmatrix} \pi^{2'}_{11} & \pi^{2'}_{12} \\ \pi^{2'}_{21} & \pi^{2'}_{22} \end{bmatrix} = IIF \,.\, \begin{bmatrix} \pi^{2}_{11} & \pi^{2}_{12} \\ \pi^{2}_{21} & \pi^{2}_{22} \end{bmatrix} \quad \ldots \ (1)$$

$$where\ IIF = \begin{bmatrix} \frac{(1-\gamma_1)\, r_1}{(1-\gamma_1)\, r_1 + \gamma_2\, r_2} & \frac{\gamma_2\, r_2}{(1-\gamma_1)\, r_1 + \gamma_2\, r_2} \\ \frac{\gamma_1\, r_1}{\gamma_1\, r_1 + (1-\gamma_2)\, r_2} & \frac{(1-\gamma_2)\, r_2}{\gamma_1\, r_1 + (1-\gamma_2)\, r_2} \end{bmatrix}$$

We recall from Section 5.1.1 that Nash equilibrium strategies (π^1_*, π^2_*) of the players can be computed using the algorithm discussed in [28]. To reach this equilibrium the defender has to execute his/her apparent strategy $\pi^{2'}_*$after computing it using equation 1. In equation 1, he/she has to replace π^2 by π^2_*.

We now discuss the benefit of our approach compared to the perfect information model. If the defender follows the perfect information model he/she executes π^2_* as the apparent strategy. In that case, the defender ends up playing the true strategy π^2 given by the following matrix equation.

$$\pi^2 = IIF^{-1}.\, \pi^2_*$$

As a result, the true strategy π^2 deviates from the Nash equilibrium strategy when IIF is not an identity matrix. So, the equilibrium is not reached and the attacker can gain higher payoff as shown by our simulation results. Moreover, there exists such a stochastic game for which no feasible π^2 exits corresponding to the Nash equilibrium strategy, π^2_*. Some of our simulation experiments illustrate such a game.

5.1.3 Simulation

We validate the above analysis using simulation experiments as discussed below.

5.1.3.1 Simulation Framework

We simulate a stochastic game being played between an attacker and a system administrator using MATLAB. We implement an application that is able to produce the pair of optimal strategies for a zero-sum game with imperfect information. This application is based on the modified Newton's method as described under article 3.3 in [28]. An iterative non-linear optimization algorithm is used. The input to this algorithm includes the state transition matrix and the reward matrix. As this is a zero-sum game, only the first player's reward matrix is given as input.

To compute the output, the modified Newton's method requires solving a *matrix* game in each iteration. This functionality is achieved by using an additional component that generates the optimal strategies and the value for a zero-sum matrix game as in [95].

5.1.3.2 Simulation Results

We demonstrate the feasibility and effectiveness of our model by using games as discussed in Section 5.1.2.2. Figure 4 displays the two system states and the transitions possible among them. The actions possible by the attacker during either state are *a (attack)* or *na (no attack)*. The *attack* action indicates the execution of an attack with the motivation to bring the network to *HackedState* or to continue further attacking in *HackedState*. The actions possible by the defender during either state are *d (defense)* or *nd (no defense)*. The *defense* action indicates the execution of a restore process with the motivation to bring back the network to *NormalState* from the *HackedState* or to strengthen the *NormalState* by increasing the monitoring level. The *na* or *nd* action indicates an instance of no action. We set the discount factor β to 0.75 and defender's sensor's two error probabilities, γ_1 and γ_2 as 0.1 and 0.05, respectively.

Our first experiment shows that perfect information models [58, 59] *can give higher payoff to the attacker compared to our model.* The state transition probabilities and the reward matrices are shown in Tables 4 and 5. This style of representation is based on that in [28]. The rows for each state represent the actions possible by attacker and columns represent the actions possible by defender. Each element is divided by a diagonal into two halves where the upper half represents the reward to the attacker from that state and the lower half represents the state transition probabilities when the corresponding actions are performed by both players. For example, in *NormalState*, when the attacker and defender both perform their first actions, the reward to the attacker is 10 and the probability of the network remaining in *NormalState* is 0.7 and changing to *HackedState* is 0.3 (First row in Table 4).

Table 4: The Rewards (to the attacker) and State Transition Probabilities at *NormalState* in the first experiment

	Defender's Action 1 (d)	Defender's Action 2 (nd)
Attacker's Action 1 (a)	(10) (0.7,0.3)	(40) (0.7,0.3)
Attacker's Action 2 (na)	(200) (1,0)	(0) (1,0)

Table 5: The Rewards (to the attacker) and State Transition Probabilities at *HackedState* in the first experiment

	Defender's Action 1 (d)	Defender's Action 2 (nd)
Attacker's Action 1 (a)	(200) (0.4,0.6)	(55) (0,1)
Attacker's Action 2 (na)	(45) (0.8,0.2)	(550) (0,1)

We calculate the pair of true optimal strategies, which are optStrat1 = [0.8696 0.1304; 0.8735 0.1265] (for the attacker) and optStrat2 = [0.3557 0.6443; 0.7014 0.2986] (for the defender). The value of the game V (the attacker's payoff when the game starts from *NormalState*) is found to be 284.5637. However, since the defender's sensor is faulty, he/she cannot directly execute this true strategy. The apparent strategy for the defender is computed as appStrat2 = [0.3708 0.6292; 0.6620 0.3380] using equation (1). Apparent strategy for only the defender is considered in our example as it is assumed that only the defender is uncertain about the present state of the system and not the attacker.

Our model suggests the defender to execute the apparent strategy (appStrat2) and the value of the game (V) thus obtained is 284.5637. It is verified that in reality, the true strategy optStrat2 gets executed every time this apparent strategy is played by the defender. Therefore the Nash equilibrium is attained and the value of the game (V) remains the same as previous. Since the Nash equilibrium is attained, if the defender adheres to appStrat2, the attacker cannot gain a higher payoff than V if he alters his strategy.

If the defender were to follow a game model based on perfect information, optStrat2 would be his/her apparent strategy. This scenario was also simulated and it was observed that the game is not in Nash equilibrium. This was observed by setting the attacker's strategy to Strat1 = [0 1; 0 1] and the value of the game (V_A) obtained was 422.8347, which is higher than V. Note that the increment in the attacker's gain can be much higher depending on the specification of the particular game (e.g., reward matrices and transition probabilities). It was also verified that, if the attacker adheres to optStrat1 (which corresponds to the Nash equilibrium), then the value of the game remains the same as expected.

We now discuss our second experiment which shows the existence of such a game where strategies suggested by perfect information models could not be executed. For the game in Tables 6 and 7, the true optimal

Table 6: The Rewards (to the attacker) and State Transition Probabilities at *NormalState* in the second experiment

	Defender's Action 1 (d)	Defender's Action 2 (nd)
Attacker's Action 1 (a)	(80) (0.7,0.3)	(-20) (0.7,0.3)
Attacker's Action 2 (na)	(100) (1,0)	(0) (1,0)

Table 7: The Rewards (to the attacker) and State Transition Probabilities at *HackedState* in the second experiment

	Defender's Action 1 (d)	Defender's Action 2 (nd)
Attacker's Action 1 (a)	(300) (0.4,0.6)	(100) (0,1)
Attacker's Action 2 (na)	(300) (0.8,0.2)	(100) (0,1)

strategy obtained for defender was optStrat2 = [0 1; 1 0]. Apparent strategies for all possibilities of true strategies were calculated and it was observed that none of them were equal to optStrat2. This is illustrated by Figure 5 that shows the Euclidian distance between the calculated apparent strategy (1st column) and the true optimal strategy optStrat2 (1st column). We observe that no point in the graph touches the XY plane, which signifies that no possibility of true strategies can lead to an apparent strategy equal to optStrat2. This result shows that it is not always possible for the defender to execute the strategy (optStrat2) prescribed by the perfect information model.

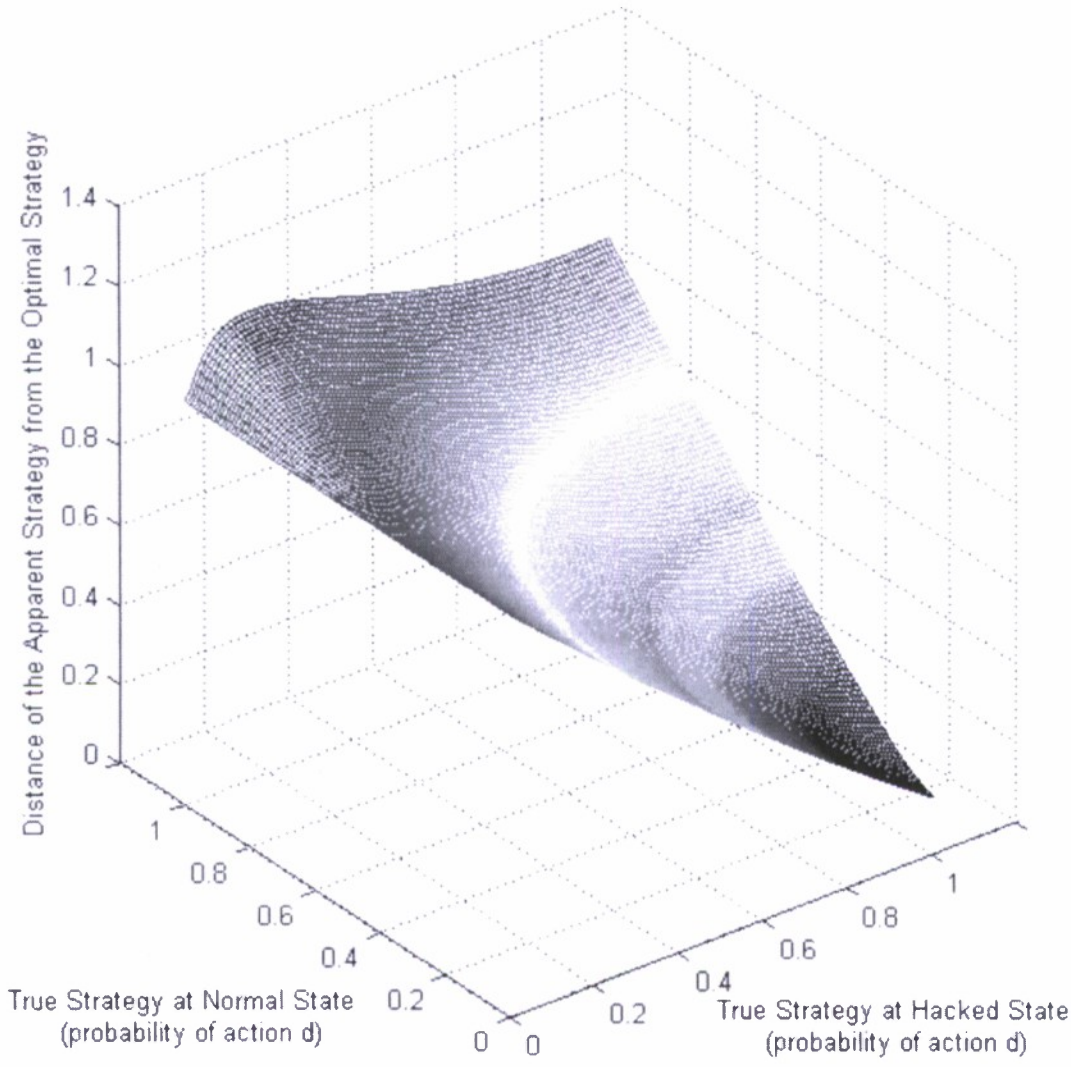

Figure 5: The second experiment result — this plot implies that the defender cannot execute an apparent strategy which is same as the *optimal strategy*

5.1.4 Related Work

The areas of game theory which are relevant to information warfare were outlined in [39]. A methodology to model the interactions between a DDoS attacker and some defense mechanism such as 'pushback' was presented in [55]. The following papers are most relevant to our work.

A perfect-information stochastic general-sum game and computed the Nash equilibrium using simulation was proposed in [58, 59]. Unfortunately, the used equilibrium computation algorithm for this general-sum

game was not available in these papers.

A general-sum, static, finite game with dynamic information was proposed in [2]. Moreover, [3] presented an imperfect information repeated game with 'finite steps' or 'infinite steps'. They analyzed the Nash equilibrium in the general-sum setting.

The operation of the IDS using a finite-state Markov chain was captured in [4]. With a few numerical examples, tools such as minimaxQ and naive Q-learning were used to find the best strategies of the players. [67] viewed the security problem as a general-sum repeated game. This model considers that the players cannot make perfect observations of each other's previous actions.

Table 8 compares our work with the prior body of research. The dimensions used for the comparison include the type of analysis (static, dynamic or none) present in the work.

Table 8: Comparing Our work with the Prior Body of Research

Work	**Stochastic game?**	**Perfect information?**	**Zero-sum / general-sum game**	**Type of analysis**
Lye et al. [58, 59]	Yes	Perfect	General-sum	Static
Alpcan et al. [2]	No(static game)	Imperfect	General-sum	Static
Alpcan et al. [3]	No(repeated game)	Imperfect	General-sum	Dynamic
Alpcan et al. [4]	Yes	Imperfect	Zero-Sum	Only Numerical Examples
Nguyen et al. [67]	No(repeated game)	Imperfect	General-Sum	Dynamic
Our work	Yes	Imperfect	Zero-sum	Static

5.1.5 Concluding Remark

Techniques that were proposed in the literature used stochastic game models to emulatenetwork security game, and showed how to determine the best strategy for the defenderconsidering the possible attack strategy used by the attacker. However, the prior research work assumed that the players have perfect information about the current state of the game, which generally does not hold in reality. Our model relaxed this assumption and enriched the prior game models by enabling them to capture more realistic scenarios. Section 5.1 presented a theoretical analysis using which the system administrator can compute his/her best strategy to reach the Nash equilibrium of a stochastic game even if the IDS sensor is imperfect. Our theoretical results were validated via simulation experiments.

Section 5.1 presented a static analysis to compute the best stationary strategy of the players. It was not discussed how the equilibrium can be reached during the game being played. We propose to investigate an

answer to this question in the future work.

5.2 Analyzing a General-Sum Stochastic Game

We have analysed paper [87] where an algorithm for computing Nash Equilibrium strategies was presented. In that paper example of general-sum stochastic game - Pollution Tax game was depicted along with Nash Equilibrium strategies found by algorithm introduced by the authors of that paper. We have implemented that algorithm in Matlab. The only difference in our algorithm is in a way of determining the initial feasible point because that part of the algorithm was not described clearly enough. Selection of the initial point can have impact on speed of the algorithm but does not have impact on the results.

In section 5.2.2 a definition of the 2-player discounted general-sum stochastic game will be presented. In the next section algorithm presented in paper [87] will be described along with our implementation of that algorithm. In section 5.2.3.6 an example coming from paper [87] is described. Results of our experiments are presented in section 5.2.3.7. In the last section we describe our concerns regarding algorithm described in paper [87] and future work which would improve that algorithm.

We need to mention here that the paper [87] that we used has not been published so far and we only have the preliminary version of it made accessible to us by the authors.

5.2.1 History

A stochastic game was introduced in the paper by Lloyd Shapley presented in 1953. Lloyd Stowell Shapley is an American mathematician and economist. He has contributed to the fields of mathematical economics and game theory.

5.2.2 Definition

We have already presented the definition of the 2-player discounted general-sum stochastic game in Section 5.1.1. To help exposition, we now present the definition with slightly different notations.

2-player discounted stochastic game is a tuple $(S, A^1, A^2, p, r^1, r^2, \beta)$, where:

- $k \in \{1, 2\}$ - the set of players
- $S = \{1, ..., N\}$ - the set of states
- $m^k : S \to \Re$ - function assigning to each state number of possible actions of k-th player in that state
- $A_s^k = \{a_1^k, ..., a_{m^k(s)}^k\}$ - set of all actions which can be taken by player k-th in the state s
- $A^k = \bigcup_{i=1}^{N} A_{s=i}^k$ - set of all actions which can be taken by k-th player
- $p : B \to [0, 1]$ where $B \subseteq S \times A^1 \times A^2 \times S$ - state transition function; $p(s^{'}|s, a^1, a^2)$ is the probability of going to state $s^{'}$ from the current state s when the player 1 chooses an action $a^1 \in A_s^1$ and the player 2 chooses an action $a^2 \in A_s^2$

- $r^k : C \to \Re$ where $C \subseteq S \times A^1 \times A^2$ - payoff function; $r^k(s, a^1, a^2)$ is the payoff of player k-th when the player 1 chooses an action $a^1 \in A^1_s$ and the player 2 chooses an action $a^2 \in A^2_s$
- $\beta \in [0, 1)$-discount factor

A decision rule f_t (resp., g_t) of player 1 (resp., 2) at time t is a function which assigns to each action the probability of taking that action at time t. The **strategy** is called **stationary** if the decision rule does not depend upon the time t it depends only on the current state of the game.[87]

The classical noncooperative assumption of game theory postulates that the players choose their strategies entirely independently (and secretly), and that they are only interested in maximizing their individual overall payoff functions. What is more, the players have precise knowledge about each other's presence in the game and payoff functions.[28]

Let f be strategy of player 1 and g be strategy of player 2, then the total expected β-discounted payoff of player k-th, is given by

$$v^k_\beta(s, f, g) = \sum_{t=0}^{\infty} \beta^t \mathbb{E}^s_{fg}(r^1_k)$$

where r^k_t denotes the reward of player k-th at time t.

We say that (f^*, g^*) is a **Nash equilibrium** (named after **John Nash** , who proposed it) if

$$v^1_\beta(s, f, g^*) \leqslant v^1_\beta(s, f^*, g^*)$$

$$v^2_\beta(s, f^*, g) \leqslant v^2_\beta(s, f^*, g^*)$$

for all strategies f, g and all states s.

Let us define the optimization problem:

$$\begin{gathered}
\min_{v^1,v^2,f,g} \sum_{k=1}^{2} 1_{|S|}^{T}[v^k - r^k(f,g) - \beta P(f,g)v^k] \\
R^1(s)g(s) + \beta \sum_{s'=1}^{N} P(s'|s)g(s)v^1(s') \leqslant v^1(s)1_{m^1(s)} \\
f(s)R^2(s) + \beta \sum_{s'=1}^{N} f(s)P(s'|s)v^2(s') \leqslant v^2(s)1_{m^2(s)}^{T} \\
\sum_{a^1=1}^{m^1(s)} f(s,a^1) = 1 \\
\sum_{a^2=1}^{m^2(s)} g(s,a^2) = 1 \\
f(s,a^1) \geqslant 0 \\
g(s,a^2) \geqslant 0
\end{gathered} \tag{1}$$

for all $a^1 \in A^1(s)$, $a^2 \in A^2(s)$ and $s \in S$.

Above optimization problem can be equivalently represented in the following way [87]:

$$\begin{gathered}
x_{jN+i} = v^{j+1}(i) \text{ for } i = 1,...,N \text{ and } j = 0,1 \\
x_{2N+m^1(0)+...+m^1(j)+i} = f(j+1,i) \text{ for } i = 1,...,m^1(j+1) \text{ and } j = 0,...,N-1 \\
x_{2N+m^1+m^2(0)+...+m^2(j)+i} = g(j+1,i) \text{ for } i = 1,...,m^2(j+1) \text{ and } j = 0,...,N-1
\end{gathered}$$

where $m^k(0) = 0$.

$$\begin{gathered}
\min_x h(x) \\
c_i(x) \leqslant 0 \text{ for all } i \in I_1 \\
c_i(x) = 0 \text{ for all } i \in E \\
c_i(x) \leqslant 0 \text{ for all } i \in I_2
\end{gathered}$$

Below we present theorems holding for general-sum stochastic games. Their proofs can be found in [28]. These theorems justify the algorithm presented in [87], for computing Nash Equilibria in general-sum stochastic games.

Theorem 1 (Filar and Vrieze [28]). *In a general sum, discounted stochastic game, there exists a Nash equilibrium in stationary strategies.*

Theorem 2 (Filar and Vrieze [28]). *Consider a point $x^{*T} = ((v^{1*})^T, (v^{2*})^T, f^*, g^{*T})$. Then the strategy part (f^*, g^*) of x^{*T} forms a Nash equilibrium of the general-sum discounted stochastic game if and only if x^* is the global minimum of the optimization problem 1 with $h(x^*) = 0$.*

Theorem 3 (Filar and Vrieze [28]). *Let* $x^{*T} = ((v^{1*})^T, (v^{2*})^T, f^*, g^{*T})$ *be feasible for 1 with an objective function value* $h(x^*) = \gamma > 0$. *Then the strategy part* (f^*, g^*) *of* x^{*T} *forms an* ϵ *- equilibrium with* $\epsilon \leqslant \frac{\gamma}{1-\beta}$.

5.2.3 Algorithm

In paper [87] was provided solution to the optimization problem 1. Their algorithm is based on Sequential Quadratic Programming. In each iteration of the algorithm 5.2.3.5 they solve a quadratic programming subproblem described in section 5.2.3.1 and defined in [87]. In section 5.2.3.2 will be presented a Hessian defined in [87] which is an input to the quadratic program.

5.2.3.1 Quadratic programming subproblem

Below we describe the quadratic programming subproblem.

Let x be a feasible point of 1 and ρ be the trust region radius. The global minimum d of the quadratic program $QP(x, \rho, H)$ given by

$$\begin{aligned} \min_d q(d) &= d^T \nabla h(x) + \frac{1}{2} d^T H(x) d \\ c_i(x) + (\nabla c_i(x))^T d &\leqslant 0, i \in I_1 \\ c_i(x) + (\nabla c_i(x))^T d &= 0, i \in E \\ c_i(x) + (\nabla c_i(x))^T d &\leqslant 0, i \in I_2 \\ \|d\|_\infty &\leqslant \rho \end{aligned} \tag{2}$$

gives the required descent direction. Matrix H(x) needs to be positive definite for the optimization problem 2 to be a convex optimization problem [11] p.152.

5.2.3.2 Hessian

In this subsection we present construction of the Hessian matrix [87].

For each $s \in S$, let

$$\lambda_s^k = (\lambda_s^{k,1}, \lambda_s^{k,2}, ..., \lambda_s^{k,m^k(s)})$$

be vectors with positive components and define

$$\tilde{\lambda} = (\lambda_1^1, \lambda_2^1, ..., \lambda_N^1, \lambda_1^2, \lambda_2^2, ..., \lambda_N^2).$$

Let $c(x) = (c_1(x), c_2(x), ..., c_{m^1+m^2}(x))^T$, where the components of the vector $c(x)$ are in the same order as in the definition of the vector $\tilde{\lambda}$.

Let

$$L(x, \lambda) = h(x) + \lambda^T c(x)$$

and

$$\begin{aligned} \lambda_s^{1,a^1} &= x_{2N+m^1(1)+m^1(2)+...+m^1(s-1)+a^1} \\ \lambda_s^{2,a^2} &= x_{2N+m^1+m^2(1)+m^2(2)+...+m^2(s-1)+a^2} \end{aligned}$$

then

$$H(x) = \nabla^2_{xx} L(x, \lambda)|_{\lambda=\tilde{\lambda}}$$

Hessian not need to be positive definite.

Let assume that A is a symmetric matrix which is not positive definite. Let $\lambda_1, ..., \lambda_n$ be the eigenvalues of A and $v_1, ..., v_n$ the corresponding eigenvectors. Consider the matrix $B = A + \mu I$, where $\mu \geqslant 0$. Then $Bv_i = (A + \mu I)v_i = Av_i + \mu I v_i = \lambda_i v_i + \mu v_i = (\lambda_i + \mu)v_i$ for all $i \in \{1, ..., n\}$. That is why when μ is large enough B is positive definite.

In the algorithm, when Hessian is not positive definite, we make it positive definite by adding absolute value of the sum of minimal eigenvalue and -0.1 to the diagonalvalues.

5.2.3.3 Initial point

In this subsection construction of the first initial point is presented.

$x^0 = ((v^1)^T, (v^2)^T, f^0, g^{0T})$, where

$$f^0(s, a^1) = \frac{1}{m^1(s)}$$

$$g^0(s, a^2) = \frac{1}{m^2(s)}$$

for all $a^1 \in A^1(s)$, $a^2 \in A^2(s)$ and $s \in S$ and v^1, v^2 are solutions of the below linear optimization problems respectively.

$$\min_{v^1, f^0, g^0} 1^T_{|S|}[v^1 - r^1(f, g) - \beta P(f, g)v^1]$$

$$R^1(s)g^0(s) + \beta \sum_{s'=1}^{N} P(s'|s)g^0(s)v^1(s') \leqslant v^1(s)1_{m^1(s)}$$

$$\min_{v^1, v^2, f^0, g^0} 1^T_{|S|}[v^2 - r^2(f, g) - \beta P(f, g)v^2]$$

$$f^0(s)R^2(s) + \beta \sum_{s'=1}^{N} f^0(s)P(s'|s)v^2(s') \leqslant v^2(s)1^T_{m^2(s)}$$

5.2.3.4 Numerical differentiation

In that section possible approximations of the derivatives will be presented. Those approximations are used in the numerical implementations of gradient and Hessian matrix of a function with n independent variables.

First derivative of a function f can be approximated in the following way:

$$f'(x) \approx \frac{f(x - h) + f(x + h)}{2h}$$

In this case the greatest accuracy is achieved when $h \approx x\epsilon^{\frac{1}{3}}$ and then the truncation error is $\epsilon^{\frac{2}{3}}$. So, with this centered diference formula we can obtain accuracy, to about the 2/3 of the machine precision (ϵ)[paste

reference].

Second derivatives of a function f can be approximated in the following way:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

The greatest accuracy is achieved when $h \approx x\epsilon^{\frac{1}{4}}$ and then the truncation error is $\sqrt{\epsilon}$.

Mixed derivatives of a function f(x,y) can be approximated in the following way:

$$f_{xy}(x,y) \approx \frac{f(x+h,y+k) - f(x+h,y-k) - f(x-h,y+k) + f(x-h,y-k)}{4hk}$$

The greatest accuracy is achieved when $h \approx x\epsilon^{\frac{1}{3}}$ and $k \approx y\epsilon^{\frac{1}{3}}$ and then the truncation error is $\sqrt{\epsilon}$. [77, 35]

5.2.3.5 SQP Algorithm

In that subsection Sequential Quadratic Programming algorithm described in [87] will be presented. It paper [87] can be also found proofs of theorems showing that this algorithm is convergent to point holding information about Nash Equilibrium.

Algorithm 1 Calculate Nash Equilibrium

Require: $\rho \geqslant 0, \sigma, \tau \in (0,1)$

Find initial feasible point x^0 such that $c_i(x^0) < 0$ for all $i \in I_1$
$x = x^0$
loop
  Calculate Hessian $H = H(x)$, make it positive definite if necessary
  $d = QP(x, \rho, H)$
  if $\|d\|_2 \leqslant \epsilon$ **then**
    return x
  end if{ϵ is machine precision}
  $\alpha = 1$
  while $c_i(x + \alpha d) \geqslant 0$ for $i \in I_1$ **do**
    $\alpha = \tau\alpha$
  end while
  if $h(x) - h(x + \alpha d) \geqslant \sigma(q(0) - q(\alpha d))$ **then**
    $x = x + \alpha d$
  else
    Reduce the trust region radius $\rho = \frac{\rho}{2}$
  end if
end loop

In our implementation the main procedure implements algorithm described in subsection 5.2.3.5. That procedure invokes procedures: determining initial point (described in subsection 5.2.3.3), calculating numerically Hessian matrix in point of a function, solving quadratic programming subproblem (described in subsection 5.2.3). There is also used procedure calculating numerically first derivative in point of a function. Implementation is tightly coupled to example described in paper [87] but could be easily adjusted to other

examples. However, the best solution would be to design universal structures and adjust implementation appropriately so that code would not have to be modified in case of different general-sum stochastic games.

Theorem 4 ([87]). *The sequence of points generated by the algorithm converges to a Karush-Kuhn-Tucker (KKT) point of the optimization problem.*

Theorem 5 ([87]). *Every KKT point of the optimization problem is a Nash equilibrium of the stochastic game.*

We have implemented our algorithm in Matlab R2008b. We used linprog and quadprog procedures from Optimization Toolbox.

Additionally we used CVX: A system for disciplined convex programming by Michael C. Grant and Stephen P. Boyd for checking some partial results. CVX is a Matlab-based modeling system for convex optimization.

5.2.3.6 Example

The following example comes from paper [87]. We could also come up with lots of other examples of general-sum stochastic games, like those related to network security (game between attacker and defender taking place in network).

Two firms produce the same product and compete for the same market. Whatever is produced is consumed. The firms contribute to the emission of a certain pollutant. More production implies higher level of pollution. Suppose the government wants to control the pollution level, but it can only detect the combined emissions. To control the emissions, the goverment imposes the same tax on both the firms.

The government imposes:

- no tax if both the firms do not pollute
- tax 2 if the pollution is at intermediate level i.e., in $(0, 4]$
- tax 4 if the pollution level is high i.e., more than 4.

So, the set of states is $S \in \{0, 2, 4\}$

The actions of the firms are the level of the pollution that they cause.
Action set for Firm 1 at state 0 - $A^1(0) = \{0, 3, 5\}$
Action set for Firm 1 at state 2 - $A^1(2) = \{0, 2, 3\}$
Action set for Firm 1 at state 4 - $A^1(4) = \{0, 2\}$
Action set for Firm 2 at state 0 - $A^2(0) = \{0, 3, 5\}$
Action set for Firm 2 at state 2 - $A^2(2) = \{0, 2, 4\}$
Action set for Firm 2 at state 4 - $A^2(4) = \{0, 3\}$

Transition probabilities and payoffs of both players are presented in Tables 9, 10, and 11.

Results presented in the paper [87] are presented in Table 12.

Table 9: Transition probabilities and payoffs of both players at $s = 0$

Firm 2 \ Firm 1	0	3	5
0	(1,0,0) (5.00, 6.00)	(0,1,0) (4.40, 7.00)	(0,0,1) (4.00, 7.67)
3	(0,1,0) (5.75, 5.40)	(0,0,1) (5.15, 6.40)	(0,0,1) (4.75, 7.07)
5	(0,0,1) (6.25, 5.00)	(0,0,1) (5.65, 6.00)	(0,0,1) (5.25, 6.67)

Table 10: Transition probabilities and payoffs of both players at $s = 2$

Firm 2 \ Firm 1	0	2	4
0	(1,0,0) (3.00, 4.00)	(0,1,0) (2.60, 4.67)	(0,1,0) (2.20, 5.34)
2	(0,1,0) (3.50, 3.60)	(0,1,0) (3.10, 4.27)	(0,0,1) (2.70, 4.94)
3	(0,1,0) (3.75, 3.40)	(0,0,1) (3.35, 4.07)	(0,0,1) (2.95, 4.71)

Table 11: Transition probabilities and payoffs of both players $s = 4$

Firm 2 \ Firm 1	0	2
0	(1,0,0) (1.00, 2.00)	(0,1,0) (0.40, 3.00)
3	(0,1,0) (1.50, 1.60)	(0,0,1) (0.90, 2.60)

Table 12: Nash Equilibria for various discount factors: These results were presented in [87].

Discount factor	Player	s=0	s=2	s=4
$\beta = 0.2$	Firm 1	(0, 0, 1)	(0, 1, 0)	(1, 0)
	Firm 2	(0, 0, 1)	(0, 1, 0)	(0, 1)
$\beta = 0.3$	Firm 1	(0, 0, 1)	(0, 0.4842, 0.5158)	(1, 0)
	Firm 2	(0, 0, 1)	(0, 0.4, 0.6)	(0, 1)
$\beta = 0.5$	Firm 1	(0.9, 0, 0.1)	(0.5473, 0.4527, 0)	(1, 0)
	Firm 2	(0.75, 0, 0.25)	(0.4802, 0.5198, 0)	(1, 0)
$\beta \in \{0.7, 0.8, 0.9, 0.98\}$	Firm 1	(1, 0, 0)	(1, 0, 0)	(1, 0)
	Firm 2	(1, 0, 0)	(1, 0, 0)	(1, 0)

For high discount factor, both firms do not pollute. Thus the government can achieve its goal of less pollution.

5.2.3.7 Empirical results

Tests were run on Laptop HP with Intel Dual Core Processor, 2100MHz, 4GB RAM, Vista64bit.

Problem 1. *For discount factors* $\beta \in \{0.2, 0.3, 0.5, 0.7, 0.8, 0.9, 0.98\}$ *and starting feasible point computed according to section 5.2.3.3 algorithm was converging too slowly.*

Problem 2. *Results in Table 13 were achieved after setting the initial point's strategies directly to those achieved by the authors of the paper [87].*

Table 13: Results from our Experiment: Nash Equilibria for various discount factors.

Discount factor	Player	s=0	s=2	s=4	h(x)	Execution time
$\beta = 0.2$	Firm 1	(0, 0, 1)	(0, 1, 0)	(1, 0)		
	Firm 2	(0, 0, 1)	(0, 1, 0)	(0, 1)	*	
$\beta = 0.3$	Firm 1	(0, 0, 1)	(0, 0.4842, 0.5158)	(1, 0)		
	Firm 2	(0, 0, 1)	(0, 0.4, 0.6)	(0, 1)	*	
$\beta = 0.5$	Firm 1	(0.9, 0, 0.1)	(0.5473, 0.4527, 0)	(1, 0)		
	Firm 2	(0.75, 0, 0.25)	(0.4802, 0.5198, 0)	(1, 0)	*	
$\beta = 0.7$	Firm 1	(1, 0, 0)	(1, 0, 0)	(1, 0)		
	Firm 2	(1, 0, 0)	(1, 0, 0)	(1, 0)	1.42e-014	64.123064 s
$\beta = 0.8$	Firm 1	(1, 0, 0)	(1, 0, 0)	(1, 0)		
	Firm 2	(1, 0, 0)	(1, 0, 0)	(1, 0)	2.13e-014	63.187655 s
$\beta = 0.9$	Firm 1	(1, 0, 0)	(1, 0, 0)	(1, 0)		
	Firm 2	(1, 0, 0)	(1, 0, 0)	(1, 0)	4.26e-014	64.586358 s
$\beta = 0.98$	Firm 1	(1, 0, 0)	(1, 0, 0)	(1, 0)		
	Firm 2	(1, 0, 0)	(1, 0, 0)	(1, 0)	4.55e-013	59.479102 s

* *To slowly convergent, but confirmed that results presented in paper [87] are wrong according to our algorithm.*

The results we obtained confirm the results obtained in paper [87] only partially.

Problem 3. *When f is set to* $[00101010]^T$ *and g is set to* $[00101001]^T$ *then after few seconds x has value x = [5.485 3.8745 1.1749 7.5008 5.7708 4.1541 0 0 1 0 1 0 0.9998 0.0002 0 0 1 -0 0.9996 0.0004 -0 1]. We can see that strategy parts are convergent to different values.*

When f is set to $[00101001]^T$ *and g is set to* $[00100101]^T$ *then after few seconds x has value x=[5.4750 3.1859 1.1250 7.3200 5.5829 3.2500 0 0 1 -0 0.9643 0.03569 0 1 0 0 1 -0 0.0273 0.9727 -0 1] We can see that strategy parts are convergent to different values.*

When f is set to $[00100101]^T$ *and g is set to* $[00100101]^T$ *Then after 68.948797 seconds we can see that* $h(x) = 5.662137425588298e - 015$ *so f, g form Nash Equilibrium.*

5.2.3.8 Limitations and Remarks

1 In the algorithm, procedures computing numerically gradient and Hessian matrix of a function should be properly modified. It is not fixed what value should have h (see section 5.2.3.4) when x=0. Currently in such case we set h to $\sqrt{\epsilon}$. In the paper [87] is not described how gradient and Hessian matrix of the functions are computed.

2 In paper [87] they compute value vectors of initial point by finding centers of maximal circles inscribed in polyhedrons of lincar optimization problems (described in [11] p. 416) described in section 5.2.3.3. Our computations show that polyhedrons of those linear optimization problems are not bounded. That is why there do not exists such circles.[11]

3 Value of the initial point has impact on rapidity of convcrgence, what show experiments presented in problem 13.

4 Objective function of the main optimization problem is not convex in general, although it can be convex function over the domain restricted by constraints.

5 More innovative version of Sequential Quadratic Programming algorithm is described in [66].

5.2.4 Additional Information

Below we present some additional relevant information.

- At the carly stage of our work, we concentrated on the article "A Computational Procedure for Gencral-sum Stochastic Games" by Prasad H. L., S. Bhatnagar, and N. Hemachandr available at http://aditya.csa.iisc.ernct.in/TR/2009/5/. We spent significant amount of time in order to understand it more dccply. This article assumed that for Nash equilibrium to be found it has to have some special properties. That is why algorithm proposed here is not general and works only for some special cases.

- The paper, "A Trust Region Sequential Quadratic Programming based Algorithm for computing Nash Equilibrium Strategies Of Stochastic Games" [87] describes general algorithm for computing Nash Equilibria in General-sum stochastic games. For that reason we have started working on understanding concepts described here. We have decided that we will start to implement this algorithm using Matlab. It occurred that some parts of that algorithm can be solved by Matlab core procedures.

- We wrote emails to authors of articles: "A Computational Procedure for General-sum Stochastic Games" and "A Trust Region Sequential Quadratic Programming based Algorithm for computing Nash Equilibrium Strategies Of Stochastic Games" [87] asking for code for their implementations of algorithms described in those papers and got answer that authors are still improving their implementations.

- We have watched online videolectures: "Game Theory" by Professor Ben Polak (YALE) and "Convex Optimization I" by Professor Stephen Boyd (Stanford University).

- Finally, wc have finishcd implemcnting the algorithm decribed in [87] and presented results in Section 5.2.3.7.

5.3 Summary

Prior researchers designed stochastic game models for network security assuming that players have perfect information. We designed a stochastic game model in which players may have imperfect information. Furthermore, prior works did not present an algorithm to compute the equilibrium of a general-sum stochastic game. We searched the game theory literature for such an algorithm and discovered that only recently theoreticians proposed one such algorithm. We analyzed this algorithm, verified the analytical results via simulation, and found that this algorithm has many limitations.

6 Game Theoretic Defense Mechanisms against a Class of Attacks

In this section, we focus on bandwidth depletion attacks for Denial of Service (DoS) or Distributed DoS (DDoS) where a single attacking node or multiple attacking nodes attempt to break down one or more network links by exhausting limited bandwidth. We consider the interaction between the attacker[1] and the defender (network administrator) as a two-player game and apply game theory-based countermeasures. For each of DoS and DDoS cases, we design a static game, which is a one-shot game where no player is allowed to change the strategy. The attacker attempts to find the most effective sending rate or botnet size while the defender's challenge is to determine optimal firewall settings to block rogue traffics while allowing legitimate ones. We study the existence of the Nash equilibrium, which represents the best strategy of each player. We also show the benefit of using the game-theoretic defense mechanisms to the network administrator. Furthermore, we present a dynamic game model that allows each player to change the strategy during the game.

We validate our analytical results through extensive simulation-based experiments in NS-3. Our simulations provide performance measurements from situations involving a single attacking node and multiple ones. We develop a new module in NS-3, NetHook, which enables an application or a module to have direct access to packets as it traverses the network stack. The addition of this module satisfies the requirements of our packet filtering specifications. More importantly, NetHook facilitates packet inspection at any arbitrary level of the NS-3 network stack and can be used to implement any of the myriad of filtering applications to provide features such as firewall, network address translation, and intrusion detection system. We also develop two additional modules based on NetHook in NS-3: NetHookFlowMon, a layer-2 flow monitoring module that provides a per-flow association of packet flow information, and NetHookFilter, a layer-3 module that implements our game-inspired filtering approaches.

6.1 Related Work

Bandwidth depletion in the form of DoS or DDoS is one of the most common attacks in cyberspace and various defense mechanisms have been proposed to mitigate the effect of such attacks [61]. We provide below a survey of related efforts.

The key of DoS/DDoS defense approaches is to identify malicious nodes and restrict their packet injection from the source or drop unwanted packets at intermediate routers before they reach the destination. PATRICIA [93] allows edge networks to cooperate to prevent misbehaving sources from flooding traffic. Lau *et. al* [50] conducted simulation-based analysis on various queuing algorithms including DropTail, Fair Queuing, Stochastic Fair Queuing, Deficit Round Robin, Random Early Detection, and Class-based queuing to determine the best queuing strategy in the target router during a DDoS attack. Chertov *et. al* [21] pointed out that DoS can be caused not only by flooding but also by exploiting the congestion window of TCP in the communication between the server and the client. Their experiments were based on the assumption that the length of the attack pulse controls the tradeoff between attack damage and attack stealthiness. During

[1] We assume that a single attacker controls all of the attacking nodes present in a botnet for DDoS.

the congestion avoidance phase, when packet losses occur, TCP halves its congestion window, which is exploited for attack.

Andersen *et. al* [6] proposed a proactive protection scheme against DDoS attacks by imposing an overhead on all transactions to actively prevent attacks from reaching the server. Their architecture generalizes the Secure Overlay Services (SOS) to choose a particular overlay routing and the set of overlay nodes are used to distinguish legitimate traffic from the attack traffic. Yaar *et. al* [100] proposed a Stateless Internet Flow Filter (SIFF) to mitigate DDoS flooding attacks based on per-flow states by protecting privileged flows from unprivileged ones. They used a handshake mechanism to establish a privileged flow that consists of marked packets with the "capability" obtained by the handshake. Wu *et. al* [96] constructed an adaptive cyber security monitoring system that integrates a number of component techniques such as decision fusion-based intrusion detection, correlation computation of event indicators, random matrix theory-based network representation of security events, and event identification through network similarity measurements.

Game theory has been widely applied in various application domains and is attracting more attention from network researchers for cyber security. Xu *et. al* [99] proposed a game-theoretic model to protect a web service from DoS attack. Network attacks [100, 81, 69, 51] have been extensively studied via simulations, which often require realistic parameters of simulated components. Our work focuses on mitigating DoS/DDoS attacks using a game theoretic approach and validating the game models in NS-3. Different from other simulation efforts, we develop several new modules of NS-3 for gathering packet statistics and mitigating malicious flows.

6.2 Network Topology

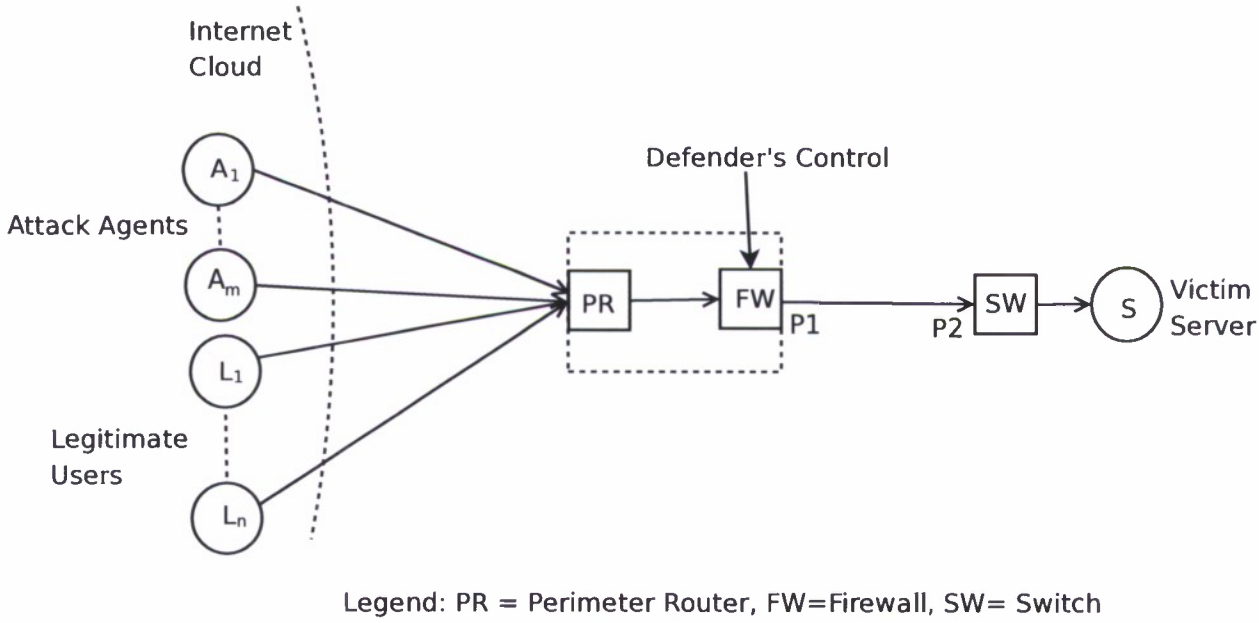

Figure 6: A generic network topology for DoS/DDoS attack.

We consider a generic network topology for DoS/DDoS attacks as shown in Figure 6, where the server S is connected to the Internet cloud via an edge switch (SW), a firewall (FW), and a perimeter router (PR). The bandwidth of the pipe (P1, P2) between the FW and the SW is limited and is subject to a DoS/DDoS attack. The defender's control is present at the FW. There are n legitimate users who need to communicate with the server S, and also, there is one attacker A who attempts to launch a denial of service attack by consuming most of the bandwidth of the pipe (P_1, P_2). The attacker A controls m attacking nodes that can send bogus packets. Note that DoS attack is a special case of DDoS attack when $m = 1$.

We would like to point out that our models and simulation-based experiments are not network-specific and are readily applicable to any DoS/DDoS scenarios in an arbitrary network topology with the following assumptions on network settings:

- A single attacker controls all of the attacking nodes, each of which sends a flow of bogus packets to the server S.
- There is an infinitely high bandwidth available on the channel between the PR and the FW, which is able to process all of the incoming packets.
- The defender has no knowledge of whether the flow is coming from the attacker or a legitimate user.
- The FW's belief on the legitimacy of the flow decreases with the increase of the flow rate.
- Some packets of a flow f are dropped in one of the two places: (i) at the FW; and (ii) at point P1 when the total incoming flow rate T is more than the available bandwidth B.
- The attacker does not spoof a unique source address for each packet in a single flow. Such spoofing would be extremely difficult and is highly unlikely to occur. Note that when the spoofed source address is the same for the entire flow, the filtering mechanism would act the same as if there were no spoofing.

For convenience, we tabulate all the notations and abbreviations used in our mathematical models in Table 14.

6.3 Game Models

In this section, we present our game models for DoS/DDoS attacks and their possible countermeasures. We consider the interaction between the attacker and the defender (network administrator) as a two-player game. We study the existence of an equilibrium in these games and also show the benefit of using the game-theoretic defense mechanisms.

The attacker attempts to find the most effective packet sending rate or botnet size, and the defender's challenge is to determine the best firewall settings to block rogue traffics while allowing legitimate ones. We first discuss some basic concepts of game theory and the profile of legitimate users, and then construct our game models.

In a *game*, each player chooses actions that result in the best possible rewards for self, while anticipating the rational actions from other players. A *strategy* for a player is a complete plan of actions in all possible situations throughout the game. A Nash equilibrium is a solution concept that describes a steady state

Table 14: Notations and abbreviations used in the models.

Symbol	Meaning
S	the victim server
PR	the perimeter router
FW	the firewall
SW	the switch
P_1	the outgoing point from FW
P_2	the incoming point to SW
B	the bandwidth of the pipe (P_1P_2) between the firewall FW and the switch SW
n	the number of legitimate users
m	the number of attacking nodes
r_l	the expected bit rate of a legitimate flow
σ_l	the standard deviation of a legitimate flow rate
r_A	the bit rate of an attack flow
γ	the minimum bit rate for a flow to be considered alive

condition of the game; no player would prefer to change his/her strategy as that would lower his/her payoffs given that all other players are adhering to the prescribed strategy.

A *static game* is a one-shot game in which each player chooses his/her plan of actions and all players' decisions are made simultaneously. A *dynamic game* is a game with multiple stages in which each player can consider his/her plan of actions not only at the beginning of the game but also at any point of time in which they have to make a decision.

6.3.1 Legitimate User Profile

We consider the presence of n legitimate users interested in communicating with the server S. The sending rate of a legitimate user is considered to be a random variable. In particular, we model the sending rate of legitimate users by picking n samples from a Normal Distribution, i.e. $X_i \sim \mathcal{N}(r_l, \sigma_l^2), i = 1, 2, ..., n$ where X_i represents the sending rate of the i-th user, r_l is the mean value of a legitimate user's sending rate, and σ_l is the standard deviation. Therefore, the total incoming flow rate with no attack is $T^{na} = X_1 + X_2 .... + X_n$. By basic laws of probability, we have $T^{na} \sim \mathcal{N}(n \cdot r_l, n \cdot \sigma_l^2)$. We assume that the pipe bandwidth B is chosen such that $T^{na} < B$ with a high probability.

We first present our static game model where one single attacker controls all of the attacking nodes. Note that there is only one attacking node in a DoS attack, while there are multiple attacking nodes in a DDoS attack. Our discussion considers m attacking nodes and is generic with respect to DoS or DDoS attacks. When m is set to be 1, we get the DoS attack scenario. We further discuss the dynamic game model highlighting its difference from the static game.

6.3.2 A Static Game

In a static game, once a player decides his/her strategy, he/she does not have a second chance to change it. We consider that the attacker's reward is not necessarily equivalent in value to the defender's cost, i.e. it could be a zero-sum or non-zero sum game. The only actions available to the attacker are to set the sending rate and to choose the number m of attacking nodes. We assume that the sending rate is the same for all of the attacking flows, which is represented by r_A. In an attack situation, the total flow rate $T = (X_1 + X_2 + ... + X_n) + m \cdot r_A$. If $T > B$, then the denial of service occurs due to a congestion condition in the pipe (P_1, P_2).

6.3.2.1 Impact of the Attack with no Defense

When there is no defence mechanism in place, all the packets of each flow pass the firewall. However, if $T > B$, only a fraction of each flow can pass through the pipe (P_1, P_2). Let α denote this fraction, which is the same for each flow. We know that $(1 - \alpha)$ fraction of each flow will be dropped at P_1: if the bit rate of a flow is r, only αr bit rate will reach the server or destination. We further assume that the bandwidth resource is shared in a fair and equitable manner, and we have $\alpha = \frac{B}{T}$. Let γ be the minimum bit rate for a flow to be considered as a flow, which depends on the specific communication protocol used, and let n_g be the average number of legitimate flows, which are able to reach the server. We get $n_g = n \cdot P[X_i > \frac{\gamma}{\alpha}]$, where n is the total number of legitimate flows and $P[X > x]$ represents the probability that the value of the random variable X is higher than x. Similarly, α fraction of each attack flow will also be dropped at P_1. So, we have the average bandwidth consumption (by the attacker) ratio calculated as:

$$v_b^{nd} = \frac{m \cdot \alpha \cdot r_a}{B} = \frac{m \cdot r_A}{n \cdot r_l + m \cdot r_A}, \tag{1}$$

and the ratio of lost users to the total number of users on average calculated as:

$$\begin{aligned} v_n^{nd} &= \frac{n - n_g}{n} \\ &= P[X_i < \frac{\gamma}{\alpha}] \\ &= P[X_i < \frac{\gamma(n \cdot r_l + m \cdot r_A)}{B}]. \end{aligned} \tag{2}$$

The attacker's objective is to increase v_b^{nd} and v_n^{nd}, which are considered as the rewards. Also, we assume that the attacker has to incur some cost to get control of an attacking node. We assume that the attacker's total cost v_c is proportional to the number of attacking nodes employed and $v_c = m$. We model the attacker's net payoff as a weighted sum of the above three quantities defined as:

$$V^a = w_b^a \cdot v_b^{nd} + w_n^a \cdot v_n^{nd} - w_c^a \cdot v_c, \tag{3}$$

where w_b^a, w_n^a, and w_c^a are the attacker's corresponding weight coefficients.

On the other hand, we model the defender's net payoff as a weighted sum defined as:

$$V^d = -w_b^d \cdot v_b^{nd} - w_n^d \cdot v_n^{nd} + w_c^d \cdot v_c, \tag{4}$$

where w_b^d, w_n^d and w_c^d are the defender's weight coefficients.

6.3.2.2 Impact of the Attack in the presence of Firewall

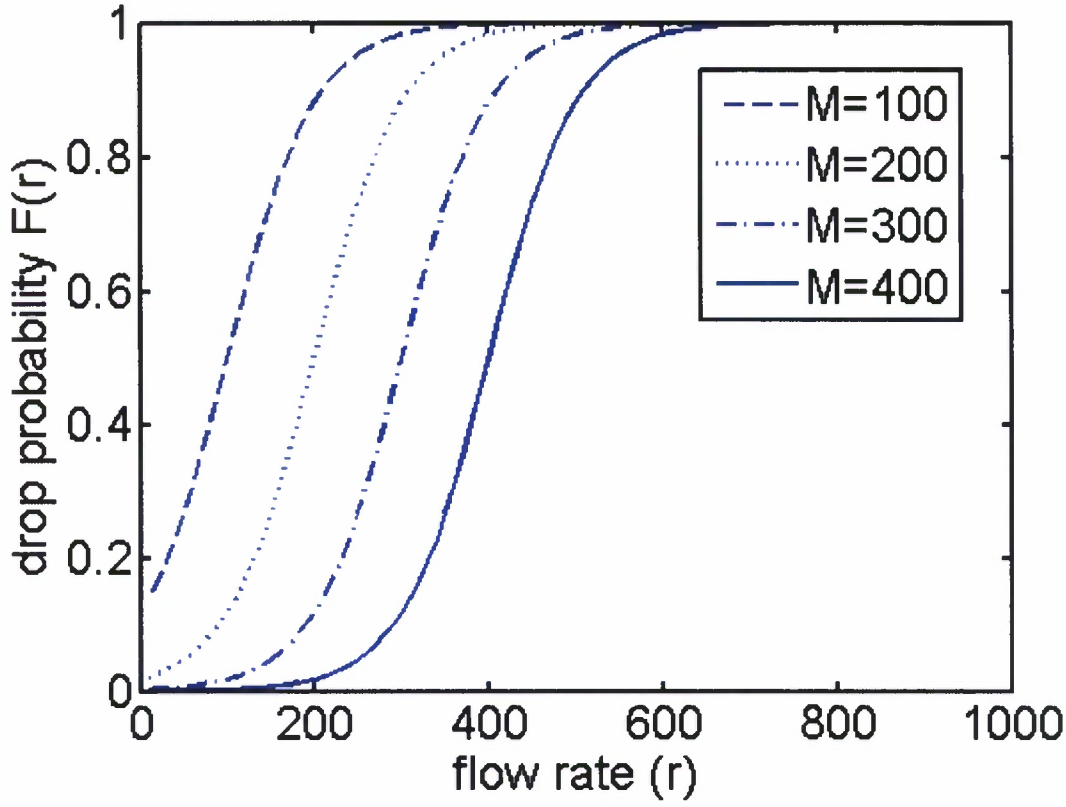

Figure 7: Plots of several sample S curves. Dropping rate of a flow at the firewall is modeled by an S curve. The X axis is the flow rate and the Y axis is the drop probability. The parameter M represents the flow rate for which the drop rate is 0.5.

The firewall is the defense agent of the network administrator: it drops the packets of an incoming flow with a probability depending on the flow rate. The dropping rate is modeled by a sigmoid function as follows:

$$F(x) = \frac{1}{(1 + e^{-\beta \frac{(x-M)}{B}})}, \tag{5}$$

where the parameter M represents the flow rate for which the drop rate is 0.5 and β is a scaling parameter. Figure 7 illustrates several sample sigmoid functions where $B = 1000$ units and $\beta = 20$. The firewall drops the packets of a flow of rate r with a probability $F(r)$. It is worth pointing out that some packets of a legitimate flow might also get dropped at the firewall. We consider that the defender controls the value of M, which is the only defense action.

Recall that r_l represents the expected rate of a legitimate flow. Let the average rate of legitimate flows passing through the firewall be r'_l. We have $r'_l = r_l \cdot (1 - F(r_l))$. On the other hand, the average rate of attacking flows passing through the firewall is $r'_A = r_A \cdot (1 - F(r_A))$. If we replace r_A by r'_A and r_l by r'_l in

Equations (1) and (2), we obtain the following results: the ratio of average bandwidth consumption by the attacker is

$$v_b = \frac{m \cdot r'_A}{n \cdot r'_l + m \cdot r'_A}, \tag{6}$$

and the ratio of lost users to the total number of users on average is

$$v_n = P[X_i < \frac{\gamma(n \cdot r'_l + m \cdot r'_A)}{B}]. \tag{7}$$

Note that the right hand side of Equation (7) considers the losses due to both the firewall and the congestion. We can compute the attacker's and defender's payoffs V^a and V^d from Equations (3) and (4), respectively, by replacing v_b^{nd} by v_b and v_n^{nd} by v_n.

We use the notion of Nash equilibrium to determine the best strategy profile of these two players. Each player has the goal to maximize his/her payoff. The attacker needs to choose optimal values for m and r_A, and the defender needs to choose the best value for M in the sigmoid function to be used by the firewall. The Nash equilibrium of this game is defined to be a pair of strategies (r_A^*, m^*, M^*), which simultaneously satisfy the following two relations:

$$V^a_{(r_A^*,\, m^*,\, M^*)} \geq V^a_{(r_A,\, m,\, M^*)} \quad \forall\, r_A,\ m$$

$$V^d_{(r_A^*,\, m^*,\, M^*)} \geq V^d_{(r_A^*,\, m^*,\, M)} \quad \forall\, M$$

We can analytically compute the Nash equilibrium strategy profile (r_A^*, m^*, M^*), which could also be obtained through numerical computation for a particular game setting. We use MATLAB as the platform for numerical computation. The following analysis shows an interesting case in which the total bytes sent by the attacker remains constant, i.e., $m \cdot r_A$ does not change, which means that the attacker only needs to set the value of m. In our future work, we will extend this analysis to a more general case. As an example, let us consider the scenario where the attacker's and the defender's weight coefficients are the same ($w_b^a = w_b^d$, $w_n^a = w_n^d$, and $w_c^a = w_c^d$), i.e., $V^a = -V^d$ (in a zero-sum game). Figure 8 illustrates the attacker's payoff V^a for different numbers m of attack flows, and different values of M with $w_b^a = 1000$, $w_n^a = 1000$, $w_c^a = 10$, $B = 2000$, $n = 20$, $r_l = 60$, $\sigma_l = 20$, $\gamma = 10$, and $m \cdot r_A = 5000$. We observe a saddle point at $m^* = 22$, $M^* = 225$, which represents the Nash equilibrium. The attack flow rate $r_A^* = 227.27$ corresponding to $m^* = 22$.

6.3.3 A Dynamic Game

In the static game model discussed above, no player has the chance to modify his/her strategy. Once the attacker sets the value for the flow rate r_A and the number m of attacking nodes, they remain fixed throughout the game. Similarly, the defender is not allowed to change the value of M, i.e. the firewall midpoint. The dynamic game model allows the players to change their strategies. This property may shift the game equilibrium point, i.e, the strategy profile (r_A, m, M) may change during the game.

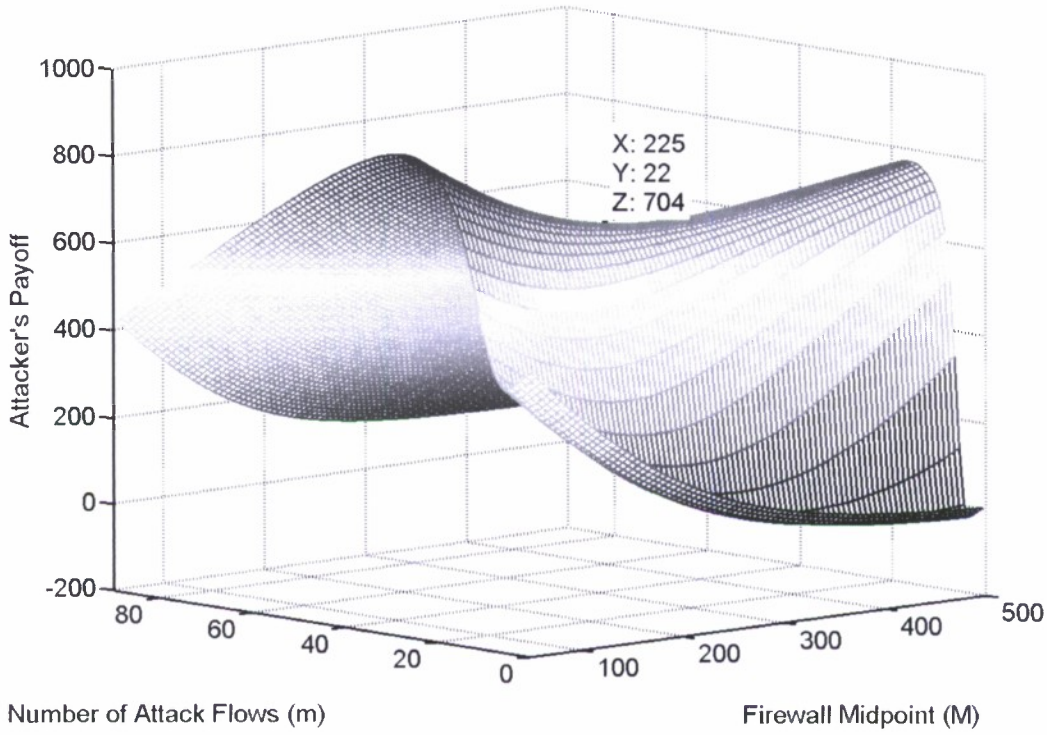

Figure 8: The attacker's payoff V^a for different numbers m of attack flows and different values of M (the firewall midpoint). We observe a saddle point at $m^* = 22$, $M^* = 225$, which represents the Nash equilibrium.

The entire game duration is considered as a sequence of time steps. As an example, the attacker A can think that if he/she sets r_A low and m high during the first few time steps, the defender D will set M to a low value, and then A can exploit it by setting r_A high and m low in the next few time steps assuming that D does not change M. On the other hand, the defender can also decide a strategy based on his/her anticipation of the attacker's behavior.

In general, it is harder to determine the Nash equilibrium for a dynamic game compared to the static game. Due to the space constraint, we do not present its formal analysis in this work. Let us consider that the game lasts for h time steps in total. When h is infinitely large, the game is said to have an infinite horizon, otherwise it is with a finite horizon.

We first extend the notations used in the static game. Let r_{A_t}, m_t, and M_t denote the corresponding quantities at the t-th time step. We represent the attacker's and defender's payoffs at the t-th time step by V_t^a and V_t^d, respectively, which are determined by the strategy profile (r_{A_t}, m_t, M_t) at that step. Similarly, the attacker's and the defender's total payoffs are denoted by V^a and V^d, respectively.

We compute the total payoff of a player by adding his/her time serial payoffs over the entire game, i.e. $V^a = \sum_{t=1}^{h} V_t^a$ and $V^d = \sum_{t=1}^{h} V_t^d$. The attacker can construct his/her strategy by deciding r_{A_t}, m_t at the t-th step $\forall t = 1, \ldots, h$. Similarly, the defender can construct his/her strategy by deciding M_t at the t-th

step $\forall t = 1, \ldots, h$. The strategy profile $(r^*_{A_t}, m^*_t,\ M^*_t,\ t = 1, \ldots, h)$ leads to the Nash equilibrium if it simultaneously satisfies the following two relations:

$$V^a_{(r^*_{A_t},\, m^*_t,\, M^*_t,\ t=1,\ldots,h)} \geq V^a_{(r_{A_t},\, m_t,\, M^*_t,\ t=1,\ldots,h)} \quad \forall\, r_{A_t},\ m_t$$

$$V^d_{(r^*_{A_t},\, m^*_t,\, M^*_t,\ t=1,\ldots,h)} \geq V^d_{(r^*_{A_t},\, m^*_t,\, M_t,\ t=1,\ldots,h)} \quad \forall\, M_t$$

6.4 Simulation

NS-3 is an advanced simulator tool written completely in C++ with optional binding for experiment scripts written in Python. There have been many recent developments with numerous research teams contributing their research as different modules for the simulator. FlowMonitor [16] is one such model which has inspired us to develop our own application to monitor packet flows. Unfortunately, FlowMonitor was not applicable in our experiment situation as it depends entirely upon the traced output of packet data, rather than inspecting these packets as they traverse NS-3's protocol stack. In our module, we need to develop a packet filtering module based on the game theory model and collect statistics on that module. For this packet-filtering module, we implement a unique network hook, which is used to observe packet flow information as they actually pass through the stack rather than at the end of the simulation.

6.4.1 Development of New Modules in NS-3

The NetHookFilter module we developed provides a means to manipulate the standard packet handling routines in NS-3. This concept has been widely used in Linux kernel for packet filtering, mangling, NAT (network address translation) and queuing packets for user-land inspection. Linux's NetFilter makes connection tracking possible through the use of various hooks in the kernel's network code. These hooks are places that kernel code, either statically built or in the form of a loadable module, can register functions to be called for specific network events at pre-defined locations within the protocol stack.

NetFilter is a useful component of modern networked systems for addressing various issues regarding packet inspection and manipulation. Traditionally NetFilter implements hooks during a packet's traversal through the protocol stack at the following locations: pre-routing, local deliver, forward, and post-routing. Each hook corresponds to locations in which one might be interested in viewing/manipulating a packet as it traverses the stack. Unfortunately, this component does not yet exist within NS-3. In order to overcome this limitation, we have developed a new NS-3 module called NetHook, which can be aggregated to any node with a network protocol stack and enables a developer to integrate their own inspection module. This new module provides the capability of manipulating packets at any location within the protocol stack.

As shown in Figure 9, NS-3 NetHook is implemented via an ordered list of callbacks associated by callback type and a given priority value. The NetHook callback list is then initiated via a call from within the existing NS-3 code via the method DoHooks(), and is capable of running any arbitrary number of hooks given the appropriate hook type. NetHook is not limited to the traditional NetFilter inspection points, pre_routing,

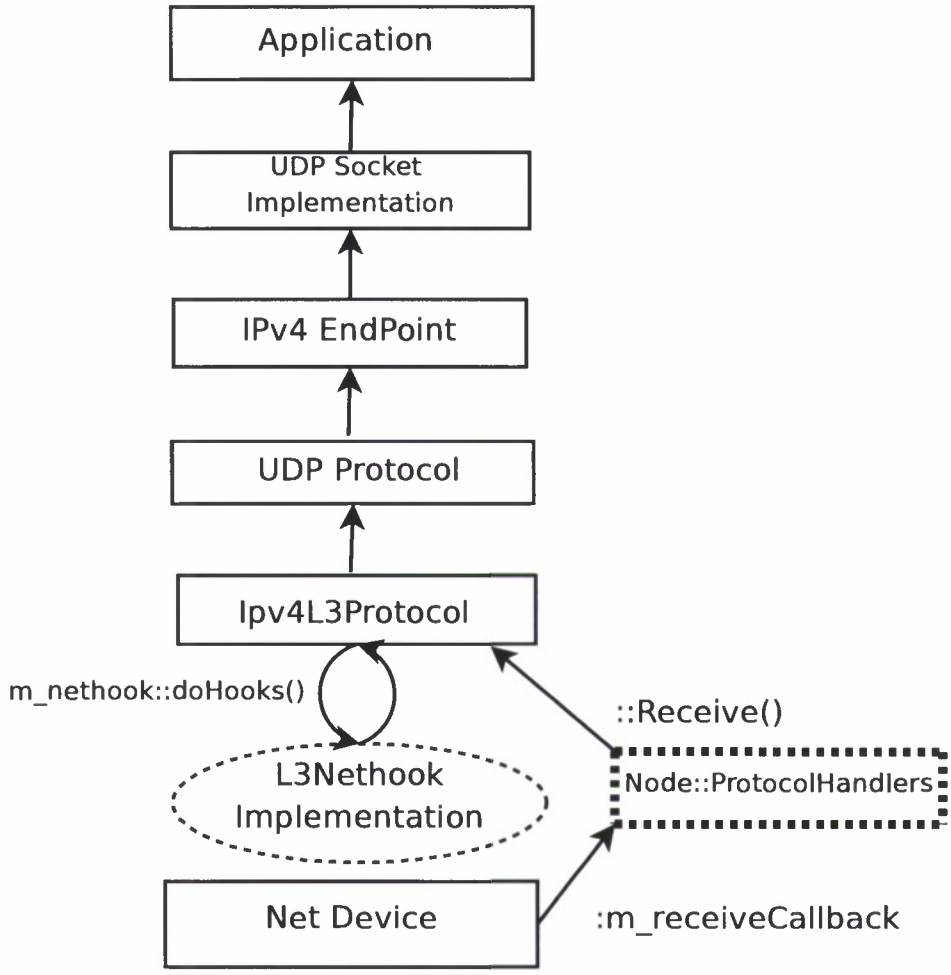

Figure 9: Implementation of NetHook. The function DoHook() enables the NetHook, which returns a boolean value that determines whether or not the packet needs to be dropped.

post_routing, local_in, local_out, and forward, rather it offers the flexibility for an NS-3 developer to implement an inspection callback at *any* location they desire within the NS-3 network system. The developer is left with the choice on the hook point for NetHook. They only need to implement the functor call and aggregate the NetHook object to the appropriate node within the topology.

6.4.2 Experimental Setup

We simulate our game-theoretic defense mechanisms in NS-3 to understand what aspects of networking would place constraints on the applicability of our model when applied to a real-world scenario. We wish to observe how control traffic would be affected, whether our model can be applied to data-intensive operations such as packet filtering, or even if the model could be applied at all. We adopt an attack model for raw bandwidth consumption where the attack nodes utilize UDP as the transport protocol in order to avoid using a modified TCP stack and avoid retransmission storms and their effect upon the simulation results. Figure 10 shows the relationship of the core infrastructure (perimeter router, firewall, and edge switch) and the packet

filtering functionality.

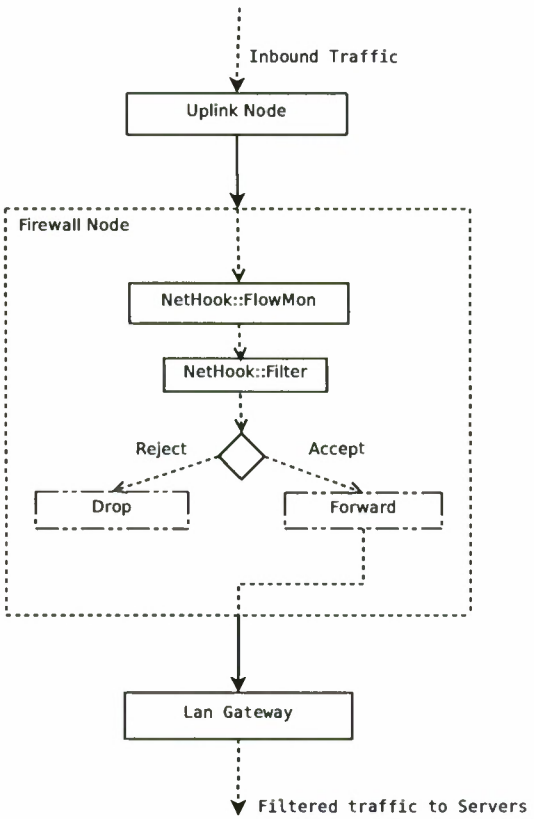

Figure 10: NetHook::Filter integration into experimental network topology.

We use the traditional dumbbell network topology for our experiments, as shown in Figure 6, which consists of three nodes where the leftmost node is the uplink node to which all legitimate and attack nodes are connected. The middle node of the dumbbell core is where we implement the packet filter. The rightmost node represents the local area network (LAN) side of the topology and provides connectivity for the server nodes. We use Point-to-Point channels to simplify the setup of the simulation topology. The left side of the topology has 1 Gbps of bandwidth while the rightmost side has 1 Mbps of bandwidth available with the bottleneck at the firewall node. The client nodes, either malicious or legitimate, are configured via the command line with arbitrary arguments for the number of nodes, packet size, and sending bit rates in order to support multiple runs with different settings. We use a constant bit rate generator available in NS-3, OnOffApplication, to generate packets destined to a server.

The experiments are run in 10 cycles, where there are 50 legitimate nodes whose packet size is 512 bytes and sends at a rate of 15Kbps. The first cycle has 5 attack nodes that send at a total of 5Mbps that is divided evenly between each attack node, and the number of attack nodes increases by 5 for each cycle. Within each cycle, we change the filter midpoint setting three times at 250Kbps, 500Kbps, and 700Kbps, respectively. Each cycle consists of 90 runs in total, 30 runs for each midpoint settings, in which there is no change in the simulation's number of attack nodes. Each run lasts for 600 seconds in length where the legitimate nodes send at a constant rate and the attack nodes begin at 30 seconds and last for 300 seconds in total. The

exact same settings are used for cases without packet filtering in order to provide a baseline performance comparison. All simulations are run on an Intel Core-2 Duo 3Ghz machine with 4Gb of RAM and running Ubuntu 9.04 with Linux kernel version 2.6.28. Each run typically takes 3-10 minutes to complete depending on the number of nodes involved. The seed value of the random number generator in each run is incremented in order to ensure independent replication of the simulation results.

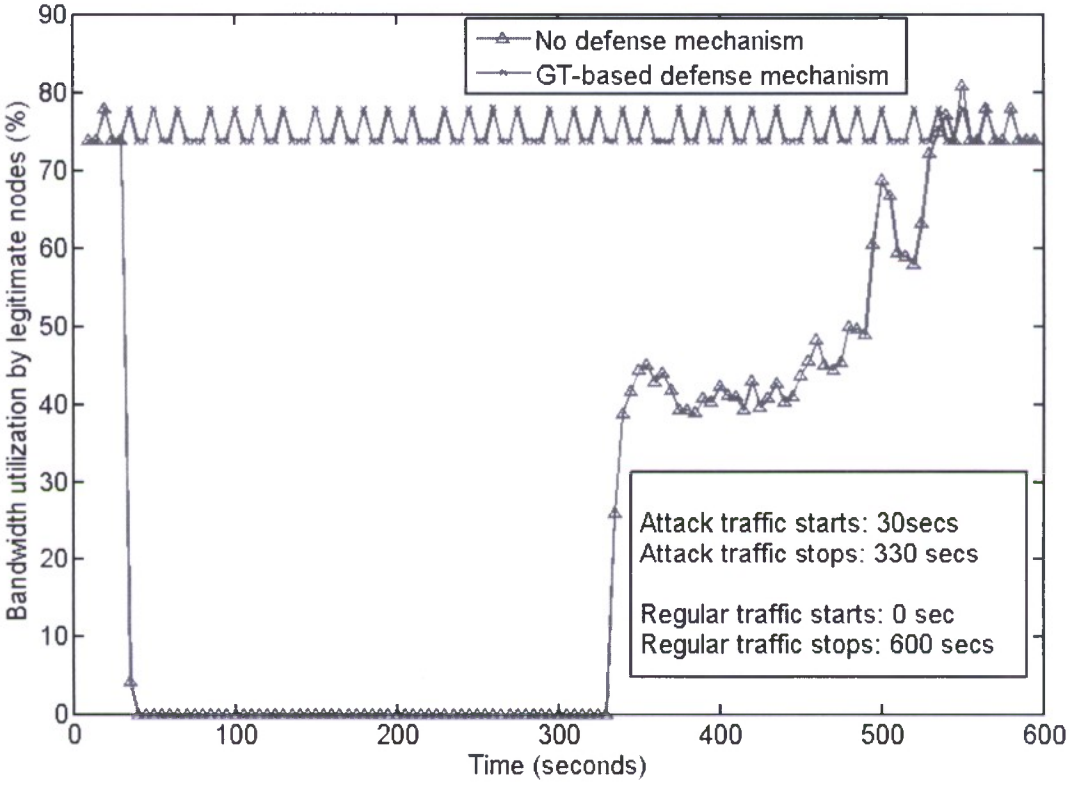

Figure 11: Impact of DDoS attack on legitimate bandwidth consumption: 5 attacking nodes transmit at 1Mbps each (total 5Mbps), 50 legitimate nodes transmit at 15Kbps each (total 750Kbps), and the S-curve midpoint is set at 500Kbps.

6.4.3 Results

The players' payoffs depend upon three components as discussed in Section 6.3 Our simulation focuses on the first component, which is the percentage of bandwidth consumed by the legitimate and attacking nodes. The second component, the fraction of active legitimate nodes, will be considered in our future work when the legitimate nodes send at different bit rates. The last component, the attacker's payoff falls outside the scope of this simulation. Figure 11 displays the effectiveness of our game theoretic defense mechanism against a DoS/DDoS attack. Figure 12 illustrates that there exists an optimal setting for the attacker, while Figure 13 shows the effectiveness of the attack can be reduced by selecting an appropriate midpoint setting. All experimental results indicate conclusively that the attacker can increase the number of attacking nodes,

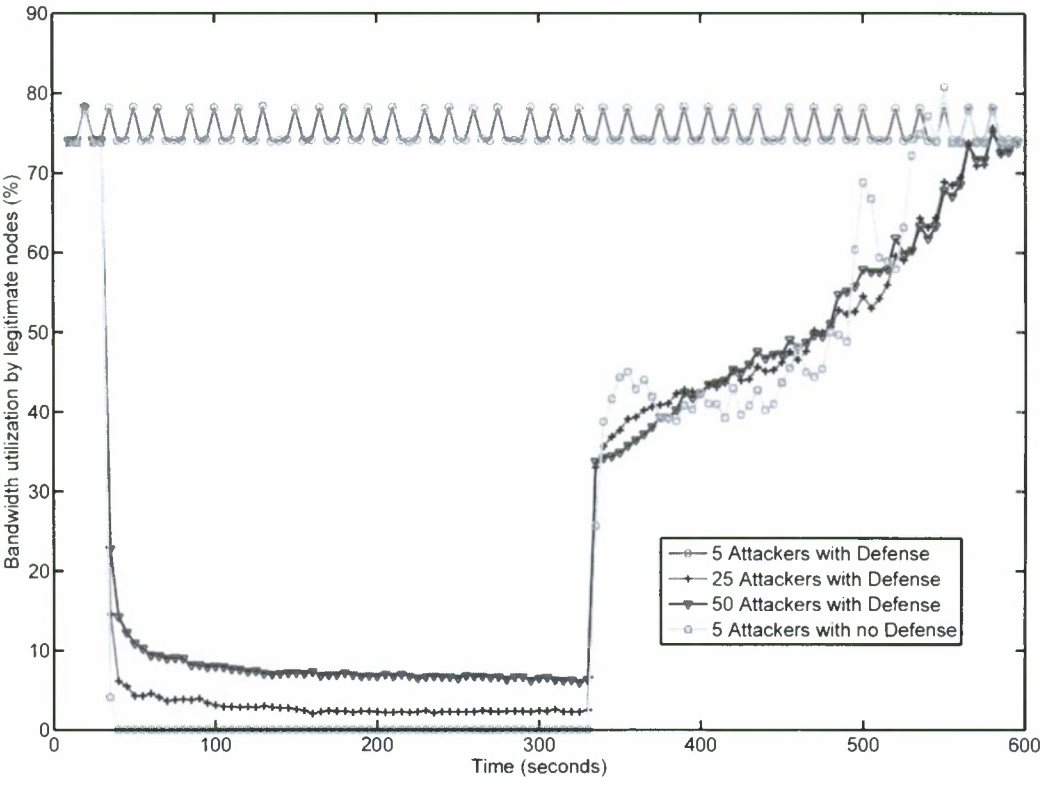

Figure 12: Bandwidth consumed by legitimate nodes when varying the number of attack nodes. The total attack bit rate remains at 5Mbps.

while decreasing the per-node bit rate, in order to bypass the filter. Conversely, the defender should select an appropriate S-curve midpoint in order to allow a majority of legitimate traffic while denying the attack traffic. If the S-curve midpoint is too high, then a large portion of the attack traffic will pass. These facts are consistent with the results from Figure 8 where we clearly see that there exists an optimal setting for both the attacker and the defender.

6.5 Summary

We presented a game theoretic model as a defense mechanism against a classic bandwidth consuming DoS/DDoS attack. Validation of our analytical results was performed utilizing the NS-3 network simulation tool.

In our future work, we will consider the existence of multiple equilibria in some scenarios. We plan to extend our simulation to incorporate a normal distribution to select the sending rate of a legitimate flow. In addition, we plan to investigate the applicability of our game-theoretic defense mechanisms in scenarios where the attacker is interested in exploiting specific protocol mechanisms to create attacking conditions. The TCP congestion window is one example of such possibilities. Furthermore, we plan to simulate a dynamic game

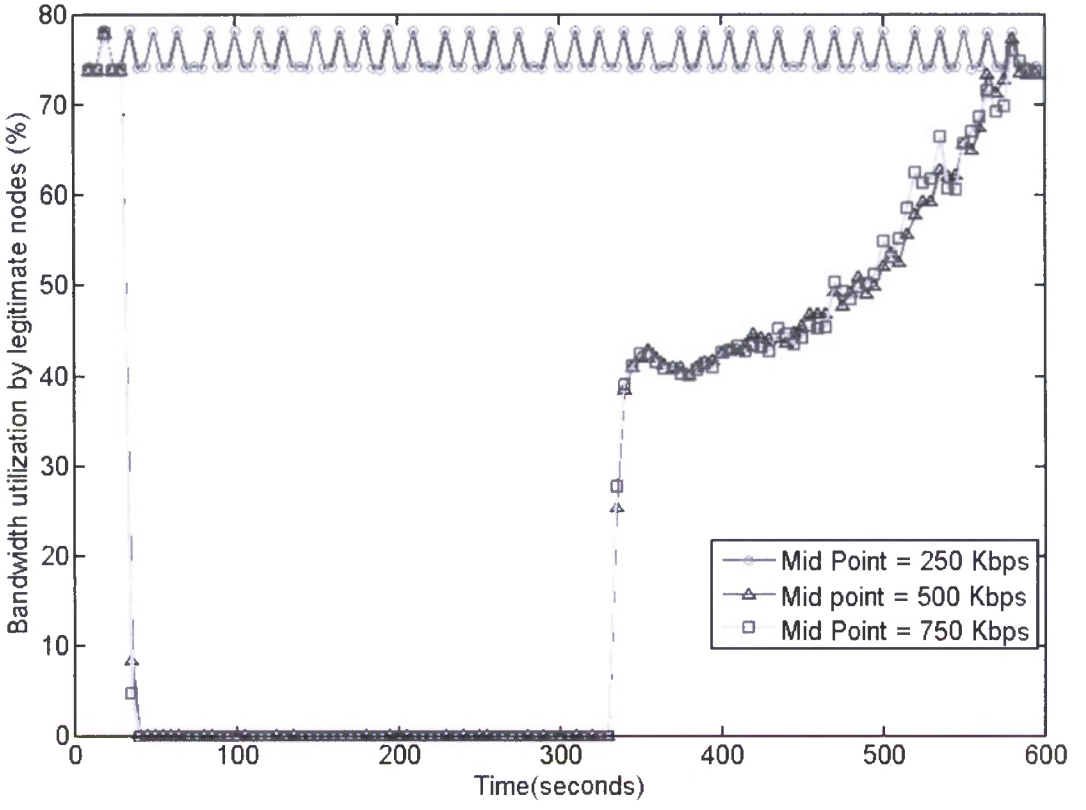

Figure 13: Bandwidth consumed by legitimate nodes when varying the S-curve midpoint. There are 15 attacking nodes whose aggregate rate is 5Mbps.

where both the attacker and the defender can alter their strategies during the attack event. We also plan to contribute our NetHook module to the NS-3 codebase in order to make it available to other researchers interested in packet manipulation within the simulator.

7 Metrics to Evaluate Game Theoretic Defense Solutions

The use of game theory approach in the network security field has increased recently. Game theory has the advantage of modeling the interactions between an attacker and defender, where players have the ability to analyze other player's behavior. This may provide the ability for an administrator to develop a better strategic defense for the system. Cronin [25] stated that a committed resource must be able to defend information warfare strategies in an effort to protect themselves against theft, disruption, distortion, denial of service, or destruction of sensitive information assets. It is vital to provide a network administrator the capability to compare different strategies using the appropriate metrics to maximize the defense of a network.

A variety of game theoretic models have been proposed with detailed information for analysis. Bellovin [7] infers that designing proper metrics to provide security measurement is a tough problem that should not be underestimated. Current research is lacking in terms of providing information which a system administrator can use in determining what type of metrics to use when developing a specific game theoretic defense model. One of the problems faced by the researchers in security games is how to evaluate different network security game models, in terms of performance, accuracy and effectiveness. The Institute for Information and Infrastructure Protection (I3P) has identified a security metrics as priority for current research and development [32]. We will extend this notion to provide a comprehensive research to disseminate metrics that will aid in analyzing the overall performance and quality of a game theoretic model. Thus, the objective of our work is to define metrics that allow the comparison of different game theoretical attack-defense models at various times. Values of these metrics may change over time due to the change of the network setting; the network configuration may change because of a part of the system being compromised or recovered [79]. Changing values in a network setting may cause a change in the level of security. We propose a taxonomy that provides a metric centric approach to disseminate vital information to evaluate the security level, performance, and quality in a game theoretic defense architecture.

Section 7 is organized as follows: In section 7.1, we provide a literature review of research that has been conducted involving security metrics. In section 7.2, we define the collected metrics and propose a few metrics. In section7.3, we discuss our idea on comparing different game theory models using the proposed metrics. In the section 7.4, we conclude this section of the report with a summary and provide insight for future work.

7.1 Related Work

In this section we will review the literature related to metrics for game theoretic defense analysis. We will also highlight literature involving metrics for risk assessment and computer information security.

He et al. [41] presented a novel Game Theoretical Attack-Defense Model (GTADM) based on network security risk assessment. This model is used to consider a threat probability that influences the attacker's decision which depends on the defender's one. The game described in this section is a non-zero sum static game with complete information. He defined an approach to formulate the payoff matrix based on the cost-benefit analysis, where many metrics are defined. These metrics include expected loss by attack, recovery by

restore, operational cost, response cost, response negative cost, expected income by attack, punishment after being detected, and cost of obtaining attack tools. Based on these metrics, the model produces the threat probability, which is the main goal of this work. Finally, this work demonstrated the advantage of using game theory in risk assessment that is taking into consideration the dependency relationship in decisions between the attacker and the defender.

Carin et al. [15] proposed a novel approach named Quantitative Evaluation of Risk for Investment Efficient Strategies (QuERIES) to quantitative cyber risk assessment. This method focused more on protecting critical Department of Defense Intellectual Property (IP). QuERIES approach involves three main elements. First, modeling the security strategy by developing an attack/protect model in a game theoretic context. Second, is to reverse engineer methodologies to develop attack graphs used for modeling attacks. Third, quantifying parameters that have been used in security strategy and attack models to evaluate and quantify the impact of the attacker's strategies on protection effectiveness. This last element performs a cost benefit analysis. For this purpose, QuERIES methodology produces relevant quantitative metrics such as the expected cost of defeating the protection, the expected time to defeat the protection, the expected cost of defeating the protection given that the protection is defeated at or before a given time, the optimal decision time for an attacker to quit if they have not succeeded, and the associated probabilities of success. All calculations of these metrics are based on the probability distribution of time (in man hours) required to successfully reverse engineer protected software.

Cremoni and Nizovtsev [24] suggested that understanding the behavior of an attacker has a big impact on the security measure. This work provided an economic model that models the behavior of the attackers when they are able to obtain complete information about the security characteristics of targets when the information is unavailable. They perform a cost benefit analysis of the attacker in order to evaluate the security level of the system. For this purpose, they proposed many metrics, such as the attacker's cost, which is to be increasing in the amount of effort spent by an attacker into an attack and the amount of effort is expressed in term of time. They believe that an attacker put more effort when the security level of the system is low. Another metric is the expected benefit from an attack.

Lye [58] presented a game theoretic model for analyzing the network security. They view the interactions between the attacker and the administrator as a two player's stochastic game and construct a model for the game. This work provides metrics that quantifies the payoff matrix of the game, which may provide a mean for improving the defense strategy. The metrics present the benefit of an attacker in terms of the amount of the damage he/she does to the network. Also, they explained that the amount of recovery effort (time) required by the administrator to bring the network from state to state is a benefit for the attacker. Some attacker's costs are difficult to quantify, for example, the loss of marketing strategy information to a competitor can cause large monetary losses.

Alpcan and Basar [3] proposed a game theoretic analysis of intrusion detection in an access control environment. They provided several common metrics that were used to help identify the performance of the IDS. Using the metrics they provided, simulation was used to determine the costs and actions of the attacker and IDS.

Bloem et al. [10] proposed an intrusion response as a resource allocation problem, where the resources being used were the IDS and network administrator. They provided insightful metrics regarding the response time of an IDS and its ability to respond without the administrator's involvement. Also, they used an administrator response time metric to determine the time of effort used to compute administrator involvement after an alert from the IDS. This metric can prove beneficial in determining how well a system is able to successfully respond against attacks while minimizing the administrator's involvement.

Liu et al. [55] proposed an incentive based modeling and inference of attacker intent, objectives, and strategies. They provided several examples the compute the bandwidth at a point in time before, during, and after an attack. They specified metrics to compute the absolute impact and relative availability to determine the system degradation. These metrics are used to distinguish how well the system was able to capitalize on the attack, as well as how well the attacker was able to succeed in reducing the bandwidth.

You and Shiyoung [101] proposed a network security behavior model based on game theory. They provide a framework for assessing security using the Nash equilibrium of game theory. In assessing the security, they also provide metrics used to analyze the payoff and cost of an attacker and defender using the exposure factor, average rate of occurrence, single loss expectancy, and annual loss expectancy.

Savola [82] surveyed emerging security metrics approaches in various organizations and provided a taxonomy of metrics that were applicable to information security. His taxonomy provided a high level approach to classifying security metrics for security management involving organization, operational, and technical aspects. He also included high level classification for metrics involving security, dependability, and trust for products, systems, and services. The metrics provided are all high level, with a lack of specific metrics used for each category, but he provides a good starting point to organizations needing to begin analyzing various security metrics within their organization.

Fink et al. [29] proposed a metrics-based approach to intrusion detection system evaluation for distributed real-time systems. They provided a set of metrics to aid administrators of distributed real-time systems to select the best IDS system for their particular organization. They presented valuable information needed to successfully gather the requirements of an organization in order to capture the importance, then use the requirements to successfully measure the performance in accordance to the requirements imposed by the organization.

7.2 Proposed Metrics

This section describes the metrics we were able to retrieve from the literature and determine how various game theoretic models can be analyzed to quantify the performance. An information security measurement principle provides insight to how well a system is performing and if financial investments are beneficial. Major benefits include improving accountability for information security activities, increasing information security performance, and providing quantifiable inputs for financial commitment.

Based on the literature review, we define several metrics that quantify the attacker's and defender's benefit and cost in a game. We divide these metrics into three categories: defender metrics, attacker metrics, and

system metrics. The defender metrics are associated to the cost and benefit of detecting an attack. The attacker metrics is the cost and benefit of attacking an asset. The system metrics is used to measure the performance and quality of a game theoretic defense architecture system.

Benefit of defender

He et al. [41] uses a non-cooperative non zero-sum game with complete information. He defines the benefit of a defender based on these metrics:

1. The damage of defender when an attack action is successful (S_{Damage})
2. The damage of defender when the attack action is detected by IDS (F_{Damage})
3. Restore which causes the recovery from the attack action
4. The detection rate of IDS (p)

He provided a calculation for each metric, such as:

$S_{Damage} = Con_p \times Con_v + Int_p \times Int_v + Ava_p \times Ava_v$

Con_p, Int_p, Ava_p are the damage degrees the attack action has made on the attack object respectively in Confidentiality, Integrity and Availability, and Con_v, Int_v, Ava_v are the objects assets in Confidentiality, Integrity and Availability. These values are not constants, and they can be set by the network administrator.

F_{Damage} = ($Con_p \times Con_v + Int_p \times Int_v + Ava_p \times Ava_v$) - *Restore*

Based on these metrics we calculate the benefit of a defender, where values depend on the attacker and defender's action. We have 4 cases:

When the attacker and defender both take actions :

$$Benefit\ of\ defender = -(S_{Damage}) \times (1-p) - (F_{Damage}) \times p$$

When the attacker takes an action and the defender decides to not defend:

$$Benefit\ of\ defender = -(S_{Damage})$$

When the attacker doesn't take any action and the defender takes an action:

$$Benefit\ of\ defender = 0$$

When the attacker doesn't take any action nor the defender :

$$Benefit\ of\ defender = 0$$

Cost of defender

He et al. [41] indicated the cost of a defender consists of Operational Cost, Response Cost and Response Negative. He provided these calculations for each metric:

1. Operational Cost can be derived from risk assessment knowledge Library.

2. Response negative (A_{Cost}) = -$P_a \times Ava_v$; P_a is in [0, 1] is the damage degree to the availability of the system caused by response actions.

3. Response Cost (R_{cost}) value is derived from the Attack-defense Knowledge Library.

4. P_m is the false detection rate of IDS.

Value of defender cost depends on the strategy of defender and attacker. Thus, we have 4 situations in the game:

When the attacker and defender take both of them, actions:

Cost of defender = - (R_{cost} + $P_a \times Av_a$) $\times$ p

- When the attacker takes an action and the defender decides to not defend:

Cost of defender = 0

- When the attacker doesn't take any action and the defender takes an action:

Cost of defender = - (R_{cost} + $P_a \times Av_a$) $\times$ P_m

- When the attacker doesn't take any action nor the defender:

Cost of defender = 0

Benefit of attacker

He et al. [41] indicates that the benefit of attacker is based on the loss of defending a system. He provides this calculation:

Benefit of attacker = - k $\times$ Benefit of defender

k is in [0,1], which is the rate that transforms the defender's loss into the benefit

of attacker's. For simplicity, k can be set to 1. Thus:

Benefit of attacker = - Benefit of defender

There are 4 situations in the game:

- When the attacker and defender take both of them, actions:

Benefit of attacker = (S_{Damage}) $\times$ (1-p) + (F_{Damage}) $\times$ p

- When the attacker takes an action and the defender decides to not defend:

Benefit of attacker = S_{Damage}

- When the attacker doesn't take any action and the defender takes an action:

Benefit of attacker = 0

- When the attacker doesn't take any action nor the defender:

Benefit of attacker = 0

Cremonini and Nizovtsev [4] defined the benefit of attacker in terms of the amount of effort put by an attacker into an attack. He mentioned that the effort is measured by time. Thus the benefit of attacker is calculated using this formula:

Benefit of attacker = E (B(x)), x: the amount of effort put in attack

$$E(B(x) = \pi(x) * G$$

$\pi(x)$: probability of success of attack given the amount of effort put into attack

G: One time payoff the attacker receives in the case of successful attack

Lye [59] defines the benefit of an attacker in terms of these two metrics:

1. Amount of damage of the attack in the system
2. Amount of recovery effort (time) required by defender to bring the network from state to state

Lye et al. benefit metric is limited with the lack of parameters used for cost benefit analysis.

Cost of Attacker

Cremonini and Nizovtsev [4] defined the cost of an attack in a function to determine the amount of effort (time) by an attacker. He doesn't provide a specific calculation for this metric. He et al. [2] defines the cost of attacker based on these two metrics:

1. Cost of lunching an attack (*Act_Cost*)
2. Punishment to the attacker (*Att_pun*) is consisted of the legal loss of the attacker

The value of the cost of attacker depends on the attacker and defender's strategy. Thus, there are 4 situations in the game:

- When the attacker and defender take both of them, actions:

Cost of attacker = $Act\_Cost + p \times Att\_pun$

- When the attacker takes an action and the defender decides to not defend:

Cost of defender = *Act_Cost*

- When the attacker doesn't take any action and the defender takes an action:

Cost of defender = *0*

- When the attacker doesn't take any action nor the defender:

Cost of defender = 0

Number of rounds to reach a Nash Equilibrium

Burke [12] proposed a metric which provides the number of turns to reach a Nash Equilibrium, in order to evaluate a game theory model of information warfare, based upon the repeated games of incomplete information model. Burke stated equilibrium provides the ability to analyze a game theory model's predictive power as it shows what strategies each player should use in order to maximize utility. The Nash equilibrium of a game involves *solving* the game – finding a unique, optimal course of play for each player. Lower NORRE values specify a player has reached equilibrium quickly, demonstrating that they are playing in harmony with the model's prediction. NORRE provides the basis for determining if players have reached equilibrium and then comparing the time to reach equilibrium across subjects and treatment conditions. Each subject's pure strategy game NORRE scores will be used to compute the average NORRE. This evaluation is done in terms of accuracy and performance.

You and Shiyong [101] provided metrics that help compute the payoff and cost of an attacker and defender using the exposure factor, average rate of occurrence, single loss expectancy, and annual loss expectancy. These metrics are defined below.

Exposure Factor (EF)

EF = percentage of loss a threat would have on a specific asset.

Average Rate of Occurrence (ARO)

ARO = Number that estimates the frequency a threat is expected to occur.

Single Loss Expectancy (SLE)

$$SLE = Asset\ Value\ \times Exposure\ Factor$$

Annual loss expectancy (ALE)

$$ALE = SLE\ \times ARO$$

Carin et al. [15] proposed a novel approach using a Quantitative Evaluation of Risk for Investment Efficient Strategies (QuERIES) to cyber risk assessment. They proposed various metrics to evaluate the Attack/Protect Model. The calculation of these metrics is based on generating a probability distribution for cost, in terms of time, of successfully defeating the protections applied to critical intellectual property (IP). Below we list the metrics used to provide risk related derivative associated to the probability of attack.

The expected cost of defeating the protection

The expected cost of defeating a protection involves the cost in man hours an attacker would have exhibit to successfully defeat the protection. The probability distribution (P_r) is based on historical data of success-

fully attacking the IP.

$$\sum_{i=0}^{\infty} c_i P_r(i)$$

where ci is the cost of the *ith* man-hour in thc attack

The expected time to defeat the protection

The expected time of defeating a protection involves the hours an attacker contributes to successfully defeat the protection. Thc probability distribution (P_r) is based on historical data of successfully attacking the IP.

$$\sum_{i=0}^{\infty} i P_r(i)$$

The expected cost of defeating the protection given that the protection is defeated at or before time t

Using the expected cost of defeating the protection, we can derive the cost of defeating the protection at or bcfore thc expected time. For instance, if an attackcr is aware of the time it takes to defeat a particular asset, but they decide to purchase additional tools to defeat the protection. This additional cost to learn the tool to defeat the protection in an accelerated fashion is calculated below.

$$\frac{\sum_{i=0}^{t} c_i P_r(i)}{\sum_{i=0}^{t} P_r(i)}$$

The optimal decision time for an attacker to quit if they have not yet succeeded

Carin et al. [15] proposed the metric to calculate the attacker's optimal decision time for an attacker to quit if attack is unsuccessful is valuable. Although, they do not define the parameters used within this metric, we can deduce the optimal time being thc average time it takes to defeat the protection of an IP. If attacker's time is greater than the average time, we can conclude the chance of being captured/dctected increases, thus the optimal time to quit correlates to the expected time to defeat the IP.

Overall Game Quality

Jansen [44] stated qualitative assignments can bc used to represent quantitative measurcs of security properties (e.g., vulnerabilities found). We define a metric Overall Game Quality (OGQ) = Availability $\times$ Performance $\times$ Quality, where the game is computed based on the availability of the system (e.g. percentage of available bandwidth), the performance of the game (e.g. average NORRE), and the quality of the system (e.g. false positive rate). This mctric is based off the ovcrall equipment effectiveness, whcre game theory parameters are applied to measure the efficiency of various games [3].

$OGQ = Availability \times Performanc \times Quality$

The Center for Internet Security [33] has developed a list of metrics that help organizations assess the performance of various assets with parameters used to calculate the metric. Below are the metrics we have selected to help evaluate our game theoretic defense architecture.

Incident Rate

Incident Rate indicates the number of detected security breaches a system or asset experienced during an allotted time period. Using incident rate, with a combination of other metrics, can indicate the level of threats, effectiveness of security controls, or attack detection capabilities.

$IR = Count(Incidents)$

Mean Time to Incident Discovery

Mean-Time-To-Incident-Discovery characterizes the efficiency of detecting attacks, by computing the average elapsed time between the initial occurrence of an incident and its subsequent discovery. The MTTID metric also serves as a leading indicator of flexibility in system or administrator's ability to defend as it measures detection of attacks from known and unknown vectors.

$$\text{MTTID} = \sum (\text{Date}_{\text{ofDiscovery}} - \text{Date}_{\text{ofOccurrence}})/\text{Count}(\text{Incidents})$$

Mean Time to Incident Recovery

Mean Time to Incident Recovery measures the effectiveness of recovering from an attack. The more responsive a system or administrator is able to respond to an attack, the less impact the attack may have on the asset.

$$\text{MTTIR} = \sum (\text{Date}_{\text{ofRecovery}} - \text{Date}_{\text{ofOccurrence}})/\text{Count}(\text{Incidents})$$

Mean Time to Mitigate Vulnerability

Mean time to mitigate vulnerabilities measures the average time exhibited to mitigate identified vulnerabilities in a particular asset. This metric is an indicator of a system or administrator's ability to patch and or mitigate a known vulnerability to reduce or remediate the risk of exploitation.

$$\text{MTTMV} = \sum (\text{Date}_{\text{ofMitigation}} - \text{Date}_{\text{ofDetection}})/\text{Count}(\text{Mitigated}_{\text{Vulnerabilities}})$$

These metrics further provide the ability to determine the best metric to evaluate in various types of game theoretic models to establish the performance and quality of the selected model. To better determine which metric is appropriate for a particular game theoretic model, the next section will provide a framework used for evaluation.

7.3 Comparing Game Theoretic Defense Solutions

Quantifying the performance of a game theoretic defense model aids a network administrator's ability to optimize defense strategies. Applying performance in addition to the payoff metrics will help capture the quality of the defense system. Figure 14 depicts the game assessment procedure (GAP) used to visualize the appropriate metrics an administrator can use to assess the performance of various game theoretic defense

models. Our objective is to compare game theoretic defense solutions, where the defender can evaluate the best game defense solution via a static or dynamic method. Dynamic method involves the defender selecting a solution at a specific time, whereas the static method involves selecting a solution without factoring in time. Supporting a defender to make such decision, we provided a list of metrics used in the evaluation process. The following scheme explains the process that a defender should follow, in order to select the best game theoretic defense solution. An analysis of how the selection process is captured is presented below.

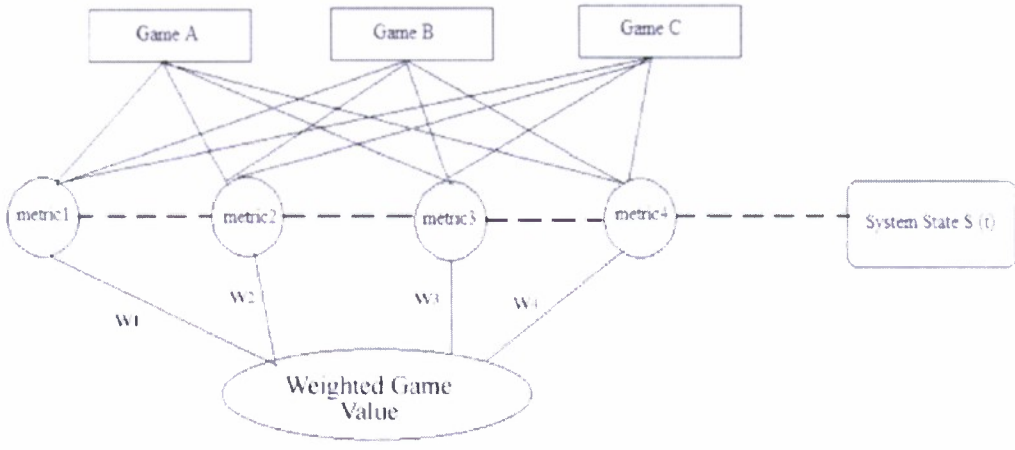

Figure 14: Game Assessment Procedure: With an example of three game models and four metrics. This proccdure is used to evaluate and select the best game for a network administrator to maximize his/her potential in defending the network

1. The defender or network administrator assigns a weight to each metric as follow: w_1, w_2, w_3, $and\ w_4$ are assigned respectively for metric1, metric2, mctric3 and metric4.

2. System State (configuration), S changes over time t. Thus, the values of metrics for a gamc are updated, which may lead to a change in the defender's game selection over time.

3. The value of each game is obtained as follows.

These calculations allow us to determine the value of each game, based on weight. The Defender's goal is to select the game which results in the highest valuc.

$$V_A(t) = \sum_{i=1}^{4} v_i(A, S(t)) * w_i$$

$$V_B(t) = \sum_{i=1}^{4} v_i(B, S(t)) * w_i$$

$$V_C(t) = \sum_{i=1}^{4} v_i(C, S(t)) * w_i$$

Figure 14 illustrates the above game assessment procedure with an example of three game models, A, B, and C and four metrics. The value of $i-$th metric at time t for game X is represented by $v_i(X, S(t))$, and $V_X(t)$ denotes the overall value of game X at time t. This procedure is used to evaluate and select the best game for a network administrator to maximize his/her potential in defending the network.

7.4 Summary

Game Theoretic models continue to provide information and analysis to initiate defense solutions against an attack for a network administrator. This section of the report is an attempt to provide an intuitive taxonomy that a network administrator can use to synthesize how well a particular game theoretic defense model is performing in a network. We list numerous metrics used in various game models. We believe providing a list of metrics for a game intrusion defense architecture will provide an administrator with the appropriate information to make an educated decision in game theoretic defense analysis. Creative metrics are evident to revolutionize a network administrator's ability to compare various defense schemes. This work provides a foundation for game theoretic defense analysis and performance evaluation.

8 Conclusion

Despite considerable effort by the research community and the practitioners for the last two decades, the cyberspace is far from completely secure. In this project we explored the usability of game theoretic defense mechanisms. We performed extensive research along a few important directions such as building a state-of-the-art attack taxonomy, a taxonomy of the existing game-theoretic solutions to cyber security problems, a stochastic game model for generic cyber activities based on realistic assumptions, a game model for the DoS/DDoS attacks and the possible countermeasures, and designing a set of metrics which can evaluate the cost and benefit of a game-theoretic defense solution. In addition, we proposed a few important research directions for future work.

We envision a semi-autonomous defense architecture which leverages a game theoretic model to counter cyber attacks. The Game Inspired Defense Architecture (GIDA) [85] will be capable of transparently observing network traffic, identifying malicious activity, measuring the risk, and acting upon that information in a way that will offer the best defense measure for that situation. The brain of GIDA is a game model which decides the best countermeasure after a thorough analysis of the cost and reward.

We stress that GIDA does not require a specific game model to be functional; any relevant game model can be plugged into GIDA. It is up to the system administrator to choose the most effective game model depending on the cyber scenario and implement the selected game model logic. The system administrator may statically selects the game model depending on his preference, network configurations, and the expected cyber scenario. We also propose to study the possibility and outcome of dynamic selection of the model during the game.

In addition to designing a game model, we will need to address a number of core research issues to realize GIDA. We envision GIDA as consisting of three key components: A set of game agents along with the central game coordinator, an administrative console, and a dynamic honeynet. These three components interact in a semi-autonomous fashion in order to provide a means to identify, evaluate, and act upon network flows. The honeynet in particular, provides a means to redirect malicious flows into dynamically instantiated honeypots for observation of malicious activity and the forensic data pertaining to it. Finally, the administrative console will provide a user interface that will allow to correlate the network state data, provide a control channel for messaging, perform forensic analysis of honeypot data, and maintain the configuration settings for the various components.

References

[1] A. Alazzawe, A. Nawaz, and M. M. Bayaraktar. Game theory and intrusion detection systems. *http://theory.stanford.edu/~iliano/courses/06S-GMU-ISA767/project/papers/alazzawe-mehmet-nawaz.pdf*, 2006.

[2] T. Alpcan and T. Basar. A game theoretic approach to decision and analysis in network intrusion detection. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 3, pages

2595–2600, 2003.

[3] T. Alpcan and T. Baser. A game theoretic analysis of intrusion detection in access control systems. *Proc. of the 43rd IEEE Conference on Decision and Control*, 2004.

[4] T. Alpcan and T. Baser. An intrusion detection game with limited observations. *Proc. of the 12th Int. Symp. on Dynamic Games and Applications*, 2006.

[5] T. Alpcan and L. Pavel. Nash equilibrium design and optimization. *International Conference on Game Theory for Networks, GameNets*, 2009.

[6] D. G. Andersen. Mayday: Distributed filtering for internet services. In *Proc. of the 4th Usenix Symposium on Internet Technologies and Systems*, March 2003.

[7] S. Bellovin. On the Brittleness of Software and the Infeasibility of Security Metrics. *IEEE Security and Privacy*, 4(4), July 2006.

[8] D. Bertsekas. Dynamic programming and optimal control. *2nd ed. Belmont, MA: Athena Scientific, vol. 2.*, 2001.

[9] M. Bishop and M. Dilger. Checking for race conditions in file accesses. *Computing Systems*, pages 131–152, Spring 1996.

[10] M. Bloem, T. Alpcan, and T. Basar. Intrusion response as a resource allocation problem. *IEEE Conference on Descision and Control*, 2006.

[11] S. Boyd and L. Vandenberghe. Convex optimization. *Cambridge University Press, New York, 2nd Edition*, 2004.

[12] D. A. Burke. Towards a game theory model of information warfare. *Air Force Institute of Technology, USA*, 1999.

[13] E. Bursztein and J. Goubalt-Larrecq. A logical framework for evaluating network resilience against faults and attacks. *Lecture Notes in Computer Science; Vol. 4846*, 2007.

[14] G. W. Bush. National strategy to secure cyberspace, office of the president. 2003.

[15] L. Carin, G. Cybenko, and J. Hughes. Quantitative evaluation of risk for investment efficient strategies in cybersecurity: The queries methodology. *IEEE Computer*, 2008.

[16] G. Carneiro, P. Fortuna, and M. Ricardo. Flowmonitor-a network monitoring framework for the network simulator 3 (ns-3). In *NSTOOLS*, Pisa, Italy, Oct. 19 2009.

[17] Cert. Avoiding social engineering and phising attacks, retrieved august 12, 2009. *http://www.us-cert.gov/cas/tips/ST04-014.html*.

[18] Cert. Denial of service attacks, retrieved august 12, 2009. *http://www.cert.org/tech tips/denial of service.html*.

[19] A. Chakrabarti and G. Manimaran. Internet infrastructure security: A taxonomy. *IEEE Network*, 16:13, November 2002.

[20] Z. Chen. Modeling and defending against internet worm attacks. *PhD Dissertation at Georgia Institute Of Technology*, 2007.

[21] R. Chertov, S. Fahmy, and N. Shroff. Emulation versus simulation: A case study of TCP-targeted denial of service attacks. In *Proc. of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, page 10, 2006.

[22] President's Information Technology Advisory Committee, Cyber Security: A crisis of prioritization, 2005.

[23] C. Cowan, P. Wagle, C. Pu, S. Beattie, and J. Walpole. Buffer overflows: Attacks and defenses for the vulnerability of the decade. In *DARPA Information Survivability Conference and Expo (DISCEX)*, pages 23–26, 2000.

[24] M. Cremonini and D. Nizovtsev. Understanding and Influencing Attackers Decisions: Implications for Security Investment Strategies. In *WEIS06: 5th Workshop on the Economics of Information Security*, 2006.

[25] B. Cronin and H. Crawford. Information warfare: Its Application in military and civilian contexts. *Information Society*, 15:257–263, 1999.

[26] C. Douligeris and A. Mitrokotsa. Ddos attacks and defense mechanisms: Classification and state-of-theart. *Comp. Networks*, 44:643–666, 2004.

[27] M. Egele, C. Kruegel, E. Kirda, H. Yin, and D. Song. Avoiding social engineering and phising attacks. In *In Proceedings of the USENIX Annual Technical Conference*, June 2007.

[28] J.A. Filar, K. Vrieze, and OJ Vrieze. *Competitive Markov decision processes*. Springer Verlag, 1997.

[29] G. Fink, B. Chappell, T. Turner, and K. Donoghue. A metrics-based approach to intrusion detection system evaluation for distributed real-time systems. In *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, April 2002.

[30] Security Focus. http://www.securityfocus.com/archive/1. *Security Focus Bugtraq Vulnerability Notification Database*, 2009.

[31] The National Strategy for Homeland Security, http://www.dhs.gov/interweb/assetlibrary/nat-strat-hls.pdf, 2002.

[32] The Institute for Information Infrastructure Protection (I3P). National Cyber Security Research and Development Challenges Related to Economics, Physical Infrastructure and Human Behavior: An Industry, Academic and Government Perspective. 2009.

[33] Center for Internet Security. The CIS security metrics. May 2009.

[34] B. Gourley. Cloud computing and cyber defense. *Crucial Point LLC*, March 2009.

[35] A. Greenbaum and T. P. Chartier. Numerical methods: Design, analysis, and computer implementation if algorithms.

[36] The Trusted Computing Group, http://www.trustedcomputinggroup.org.

[37] Aaron Hackworth. Spyware. 2009.

[38] S. N. Hamilton, W. L. Miller, A. Ott, and O. S. Saydjari. Challenges in applying game theory to the domain of information warfare. *Proceedings of the 4th Information survivability workshop (ISW-2001/2002)*, 2002.

[39] S. N. Hamilton, W. L. Miller, A. Ott, and O. S. Saydjari. The role of game theory in information warfare. *Proceedings of the 4 th information survivability workshop (ISW-2001/2002)*, 2002.

[40] S. Hansman and R. Hunt. A taxonomy of network and computer attacks. *Computers and Security*, 24:31–43, February 2005.

[41] W. He, C. Xia, H. Wang, C. Zhang, and Y. Ji. A Game Theoretical Attack-Defense Model Oriented to Network Security Risk Assessment. In *Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, volume 3, 2008.

[42] J.D. Howard and T.A. Longstaff. A common language for computer security incidents. *Sandia Report: SAND98-8667, Sandia National Laboratories, http://www. cert. org/research/taxonomy_988667. pdf.*

[43] B. Hutchinson and M. Warren. Information warfare. Corporate attack and defence in a digital world. In *Oxford: Butterworth Heinemann*, 2001.

[44] W. Jansen. Directions in Security Metrics Research. *NISTIR 7564*, March 2009.

[45] J. Jormakka and J. V. E. Molsa. Modelling information warfare as a game. *Journal of Information Warfare; Vol. 4(2)*, 2005.

[46] D. Kienzle and M. Elder. Recent worms: A survey and trends. *Proceedings of the 2003 ACM workshop on rapid malcode*, 2003.

[47] K. Killourhy, R. Maxion, and K. Tan. A defense-centric taxonomy based on attack manifestations. In *In the Proceeding of the International Conference on Dependable Systems and Networks (DSN 2004)*, June 2004.

[48] M. Kjaerland. A taxonomy and comparison of computer security incidents from the commercial and government sectors. *Computers and Security*, 25:522–538, October 2005.

[49] C.E. Landwehr, A.R. Bull, J.P. McDermott, and W.S. Choi. A taxonomy of computer program security flaws. *ACM Computing Surveys (CSUR)*, 26(3):254, 1994.

[50] F. Lau, S. Rubin, M. Smith, and L. Trajkovic. Distributed denial of service attacks. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 3, 2000.

[51] M. Liljenstam, J. Liu, D. Nicol, Y. Yuan, G. Yan, and C. Grier. Rinse: the real-time immersive network simulation environment for network security exercises. In *Workshop on Principles of Advanced and Distributed Simulation*, pages 119–128, 2005.

[52] U. Lindqvist and E. Jonsson. How to systematically classify computer security intrusions. In *IEEE Symposium on Security and Privacy*, pages 154–163. IEEE Computer Society, 1997.

[53] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. *Proc. of the 11th International Conference on Machine Learning*, pages 157–163, 1994.

[54] P. Liu, W. Zang, and M. Yu. Incentive-based modeling and inference of attacker intent, objectives, and strategies. *ACM Transactions on Information and System Security (TISSEC)*, 2005.

[55] P. Liu, W. Zang, and M. Yu. Incentive-based modeling and inference of attacker intent, objectives, and strategies. *ACM Transactions on Information and System Security (TISSEC)*, 8(1):78–118, 2005.

[56] Y. Liu, C. Comaniciu, and H. Man. A bayesian game approach for intrusion detection in wireless ad hoc networks. *ACM International Conference Proceeding Series; Vol. 199*, 2006.

[57] D.L. Lough. *A taxonomy of computer attacks with applications to wireless networks*. PhD thesis, 2001.

[58] K. Lye and J. Wing. Game strategies in network security. *Proceedings of the Foundations of Computer Security*, 2002.

[59] K. Lye and J.M. Wing. Game strategies in network security. *International Journal of Information Security*, 4(1):71–86, 2005.

[60] Microsoft. Next-generation secure computing base. www.microsoft.com/resources/ngscb/default.mspx.

[61] J. Mirkovic. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.

[62] J. Mirkovic and P. Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *Computer Communication Review 34, no. 2*, 2004.

[63] R. A. Miura-Ko, B. Yolken, N. Bambos, and J. Mitchell. Security investment games of interdependent organizations. *Proceedings of the 46th Allerton Conference*, 2008.

[64] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. *IEEE Security and Privacy no.4*, 2003.

[65] D. Moore and C. Shannon. Code-red: a case study on the spread and victims of an internet worm. In *Proceedings of the 2002 ACM SICGOMM Internet Measurement Workshop*, 2002.

[66] J. L. Morales, J. Nocedal, and Y. Wu. A sequential quadratic programming algorithm with an additional equality constrained phase. 2008.

[67] K. C. Nguyen, T. Alpcan, and T. Basar. Security games with incomplete information. *Proc. of IEEE Intl. Conf. on Communications (ICC)*, 2009.

[68] K. C. Nguyen, T. Alpcan, and T. Basar. Stochastic games for security in networks with interdependent nodes. *Proc. of Intl. Conf. on Game Theory for Networks (GameNets)*, 2009.

[69] D. Nicol, W. Sanders, and K. Trived. Model-based evaluation: From dependability to security. *IEEE Transactions on Dependable and Secure Computing*, 1(1):48–65, 2004.

[70] S. Noel, S. Jajodia, B. OBerry, and M. Jacobs. Efficient minimum-cost network hardening via exploit dependency graphs. In *19th Annual Computer Security Applications Conference (ACSAC)*, 2003.

[71] M. J. Obsborne and A. Rubinstein. A course in game theory. *MIT Press*, 1994.

[72] U.S. Department of Homeland Security, https://buildsecurityin.us-cert.gov/daisy/bsi/home.html.

[73] Director of National Intelligence. Annual threat assessment of the intelligence community for the senate armed services committee. *Statement for the Record*, March 2009.

[74] U.S. White House Office. National security presidential directive 54 / Homeland security presidential directive 23 (NSPD-54 / HSPD-23). January 2008.

[75] A. Patcha and J. Park. A game theoretic approach to modeling intrusion detection in mobile ad hoc networks. *Proceedings of the 2004 IEEE workshop on Information Assurance and Security*, 2004.

[76] P. Porras, H. Saidi, and V. Yegneswara. An analysis of conficker's logic and rendezvous points. *Malware Threat Center. SRI International Technical Report*, 2009.

[77] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. Numerical recipes in c. *Addison Wesley, Massachusetts, 2nd Edition*, 1992.

[78] N. Provos, P. Mavrommatis, M.A. Rajab, and F. Monrose. All your iframes point to us. In *Proceedings of the 17th conference on Security symposium*, pages 1–15. USENIX Association, 2008.

[79] J. Roos. Risk Management A Quantitative Approach to Information Security. *University of Twente*, 2008.

[80] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu. A survey of game theory as applied to network security. *The 43rd Hawaii International Conference on System Sciences*, 2010.

[81] C. Sarraute, F. Miranda, and J.L. Orlicki. Simulation of Computer Network Attacks. In *Argentine Symposium on Computing Technology*, Aug. 30 2007.

[82] R. Savola. A Novel Security Metrics Taxonomy for R & D Organizations. In *Proceedings of the 7th Annual Information Security Conference*, 2008.

[83] K. Scarfone and M. Souppaya. Technical guide to information security testing and assessment. *NIST (Sept. 08) http://web.nvd.nist.gov/view/vuln/detail?execution=e7s1*, 2008.

[84] S. Shiva, S. Roy, H. Bedi, D. Dasgupda, and Q. Wu. A stochastic game model with imperfect information in cyber security. *The 5th International Conference on Information Warfare and Security*, 2010.

[85] S. Shiva, S. Roy, and D. Dasgupda. Game theory for cyber security. *The 6th Annual Cyber Security and Information Intelligence Research Workshop (CSIIRW)*, 2010.

[86] S. Shiva, C. Simmons, C. Ellis, S Roy, D. Dasgupta, and Q. Wu. AVOIDIT: A cyber attack taxonomy. *Technical Report: CS-09-003, University of Memphis*, August 2009.

[87] V. V. Singh, N. Hemachandra, and K. S. M. Rao. A trust region sequential quadratic programming based algorithm for computing nash equilibrium strategies of stochastic games. June 2009.

[88] Packet Storm. http://packetstormsecurity.org/. *Packet Storm Vulnerability Database*, 2009.

[89] W. Sun, X. Kong, D. He, and X. You. Information security investment game with penalty parameter. *The 3rd International Conference on Innovative Computing Information and Control*, 2008.

[90] W. Sun, X. Kong, D. He, and X. You. Information security problem research based on game theory. *International Symposium on Publication Electronic Commerce and Security*, 2008.

[91] Gimmiv.A Trojan. Retrieved august 22, 2009. *http://www.f-secure.com/v-descs/trojan-spy-w32-gimmiv-a.shtml*.

[92] US-CERT. http://www.us-cert.gov/. *United States Computer Emergency Readiness Team*, 2009.

[93] L. Wang, Q. Wu, and Y. Liu. Design and Validation of PATRICIA for the Mitigation of Network Flooding Attacks. In *Proceedings of the 2009 International Conference on Computational Science and Engineering-Volume 02*, pages 651–658. IEEE Computer Society, 2009.

[94] C. Warrender, S. Forrest, and B. Perlmutter. Detecting intrusions using system calls: Alternative data models. *IEEE Symposium on Security and Privacy*, pages 133–145, 1999.

[95] JD Williams. The Complete Strategyst, revised edition, 1966.

[96] Q. Wu, D. Ferebee, Y. Lin, and D. Dasgupta. Monitoring security events using integrated correlation-based techniques. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, 2009.

[97] Q. Wu, S. Shiva, S. Roy, C. Ellis, and V. Datla. On modeling and simulation of game theory-based defense mechanisms against DoS and DDoS attacks. *Proceedings of the 43rd Annual Simulation Symposium (ANSS'10) in Spring Simulation Multiconference (SpringSim)*, 2010.

[98] C. Xiaolin, T. Xiaobin, Z. Yong, and X. Hongsheng. A markov game theory-based risk assessment model for network information systems. *International conference on computer science and software engineering*, 2008.

[99] J. Xu and W. Lee. Sustaining availability of web services under distributed denial of service attacks. *IEEE Transactions on Computers*, pages 195–208, 2003.

[100] A. Yaar, A. Perrig, and D. Song. Siff: A stateless internet flow filter to mitigate ddos flooding attacks. In *In Proc of IEEE Symposium on Security and Privacy*, pages 130–143, 2004.

[101] X. You and Z. Shiyong. A kind of network security behavior model based on game theory. *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2003.